Modèles de la rationalité des acteurs sociaux

Joseph El Gemayel

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Université Toulouse 1 Capitole (UT1 Capitole)

Présentée et soutenue par :

Joseph El-Gemayel

le 25 Juin 2013

Titre :

Modèles de la rationalité des acteurs sociaux

École doctorale et discipline ou spécialité :

ED MITT : Domaine STIC : Intelligence Artificielle

Unité de recherche :

Institut de Recherche en Informatique de Toulouse

Directeur(s) de Thèse :

Christophe Sibertin-Blanc
Françoise Adreit

Jury :

Alain Dutech - Chargé de Recherche INRIA - Rapporteur.
Abdel-Illah Mouaddib - Professeur, Université de Caen - Rapporteur.
Salima Hassas - Professeur, Université Claude Bernard Lyon 1 - Examinateur.
Nathalie Villa-Vialaneix - Maître de conférence, Université de Perpignan - Examinateur.
Françoise Adreit - Maître de Conférence, Université Toulouse Le Mirail - Co-Directeur.
Christophe Sibertin-Blanc - Professeur, Université de Toulouse I - Directeur

Résumé de la thèse

Le travail présenté dans ce mémoire s'inscrit dans le cadre du projet SocLab, qui propose une formalisation de la sociologie de l'action organisée de Crozier et Friedberg. Cette formalisation repose sur un méta-modèle de la structure des organisations sociales, à partir duquel il est possible de décrire la structure d'une organisation particulière, de développer une étude analytique de ses propriétés et surtout de calculer, par simulation, les comportements que les acteurs de cette organisation sont susceptibles d'adopter les uns vis-à-vis des autres.

Selon cette approche, une organisation est vue comme un système qui, en fonction du comportement des acteurs les uns envers les autres, procure à chacun d'eux une certaine capacité d'action pour atteindre ses objectifs, sans distinguer ceux qui relèvent de son rôle et ceux qui lui sont propres. Ces comportements sont relativement stabilisés, condition indispensable à la coordination des acteurs dans l'accomplissement, au moins partiel, de ce qui constitue la raison d'être de l'organisation, et donc indispensable à l'existence même de cette organisation. Ces comportements s'avèrent de plus être globalement coopératifs, facilitant ainsi la réalisation des objectifs, aussi bien ceux propres à chacun que ceux du collectif dans son ensemble.

Cette thèse porte sur la modélisation de la rationalité qui conduit un acteur social à adopter un tel comportement dans le « jeu social » que constitue un contexte d'interaction organisationnel.

Selon la sociologie de l'action organisée, cette rationalité est stratégique, guidée par la recherche de son intérêt, et elle s'exerce dans le cadre d'une rationalité (très) limitée. Le modèle proposé cherche à être vraisemblable, du point de vue social et du point de vue psycho-cognitif, et il s'inscrit dans le paradigme de l'apprentissage par renforcement. Dans la mesure où la structure de l'organisation le permet, les simulations convergent donc vers des configurations que l'on peut qualifier d'optima Pareto-équitables. On étudie aussi diverses variantes de cet algorithme correspondant à des rationalités qui conduisent une organisation à se réguler vers des configurations élitistes, protectrices ou égalitaristes, ou encore vers un équilibre de Nash.

Abstract of the thesis

The work presented in this paper is part of the project SocLab, which proposes a formalization of the sociology of organized action (Crozier et Friedberg). This formalization is based on a meta-model of the structure of social organizations, which provides means to describe the structure of a particular organization, to develop an analytical study of its properties and mainly, to calculate by simulation the behaviors that the actors of the organization are likely to adopt one to each other.

Under this approach, an organization is viewed as a system that, depending on the behavior of the actors to each other, gives every of them a certain capacity of action to achieve its objectives, without distinguishing those related to his role within the organization and those that are its own. These behaviors are relatively stable. This is an essential condition for the coordination of the actors so that they can coordinate in performing, at least partially, what constitutes the *raison d'être* of the organization. These behaviors appear also to be generally cooperative facilitating the achievement of personal objectives of each one as well as those of the collective as a whole.

This thesis focuses on the modeling of the rationality which leads a social actor to adopt such behavior in the « social game » constituted by a context of organizational interactions.

According to the sociology of organized action, this rationality is strategic, guided by the research of own interest, and it is exercised within the framework of a (very) limited rationality. The proposed model seeks to be plausible, from the social and the psycho-cognitive points of view, and it fits into the paradigm of reinforcement learning. Insofar as the structure of the organization allows it, the simulations converge towards configurations that can be described as Pareto optima. We also study variants of this algorithm corresponding to rationalities that drive an organization to regulate toward other configurations that are elitist, protective or egalitarian, or Nash equilibria.

Table des matières

Liste des figures

Liste des tableaux

Liste des algorithmes

Chapitre 1 – Introduction

Le sujet de la thèse présentée dans ce mémoire s'inscrit dans un projet interdisciplinaire de collaboration entre informaticiens et sociologues. Ce projet porte sur l'étude des organisations sociales et une théorie qui y est rattachée – la Sociologie de l'Action Organisée. Cette théorie est largement enseignée en France et utilisée par les consultants appelés pour analyser l'origine des dysfonctionnements qui pénalisent lourdement certaines organisations sociales.

Dans toute organisation sociale, on observe une relative stabilité des comportements des différents acteurs qui la composent, du moins tant que la structure de cette organisation n'évolue pas. Cette stabilité est indispensable pour qu'une organisation puisse réaliser ses objectifs, et plus généralement préserver sa raison d'être, et donc perdurer. Cependant, elle ne peut pas s'expliquer exclusivement par l'institution des rôles des acteurs, des processus, des règles etc. qui codifient les comportements des acteurs, ne serait-ce parce que ces règles sont toujours appliquées dans un contexte particulier qui nécessite leur interprétation pratique et laisse donc aux acteurs une marge de manœuvre certaine dans la façon de les mettre en œuvre.

La sociologie de l'action organisée (SAO) proposée par Crozier et Friedberg [Crozier, 1963; Crozier *et* Friedberg, 1977; Friedberg, 1993] est une théorie des organisations qui s'attache à découvrir les ressorts de cette stabilité des comportements, autrement dit à découvrir quels sont les mécanismes qui produisent cette régulation et pourquoi les acteurs se comportent comme ils le font, souvent de façon bien différente de ce qui est prescrit par l'organisation. La formalisation de cette théorie en collaboration avec des sociologues [Sibertin-Blanc *et al.*, 2006] a donné lieu au développement d'un laboratoire virtuel, *SocLab* [Mailliard, 2008].

Auparavant, nous présentons brièvement la Sociologie de l'Action Organisée (§1.1), la formalisation de la SAO et la rationalité des acteurs sociaux dans SocLab (§1.2). Ensuite, nous abordons plus spécifiquement la problématique de la thèse (§1.3). Finalement, nous résumons les différentes chapitres de ce document (§1.4).

1.1 – La Sociologie de l'Action Organisée

Depuis les années 1970, l'école française de sociologie des organisations a développé un programme de recherche dont la fécondité n'est pas discutée et qui s'attache à découvrir le fonctionnement réel d'une organisation au-delà des règles formelles qui la codifient [Crozier, 1963; Crozier *et* Friedberg, 1977; Friedberg, 1993] : la Sociologie de l'Action Organisée. Dans ce cadre, les organisations sont des « construits sociaux » contraignant le comportement des acteurs qui, en même temps, construisent l'organisation. Les acteurs ont un comportement stratégique : chacun essaie de conserver ou d'étendre son pouvoir sur les autres en agissant sur les états des ressources qu'il contrôle, afin d'avoir une certaine marge de manœuvre dans la gestion de son activité et la réalisation de ses objectifs. C'est cette finalité des comportements des acteurs qui assure leur relative stabilité. Le pouvoir d'un acteur résulte de la maîtrise de l'accès à une ou plusieurs ressources qui s'avèrent nécessaires à l'action des autres acteurs, maîtrise qui permet à un acteur de fixer les « termes de l'échange » dans ses interactions avec ceux qui ont besoin de ces ressources et de rendre son comportement plus ou moins imprévisible et ainsi préserver et/ou d'accroître son autonomie et sa capacité d'action dans l'organisation.

Chaque acteur tout à la fois contrôle et est dépendant des autres par l'intermédiaire des ressources requises pour l'action collective. Dès lors, les relations de pouvoir structurent des configurations sociales qui, relativement stabilisées, sont qualifiées de « systèmes d'action concrets » (S.A.C.). Un S.A.C. peut être réduit à deux éléments essentiels : les acteurs et les ressources. Il est donc un contexte d'interaction assez précisément délimité qui structure la

coopération d'un ensemble d'acteurs et permet la régulation de leurs comportements, de façon certes contraignante, mais sans leur ôter toute marge de manœuvre. D'après Friedberg, « tout contexte d'action peut être conceptualisé comme sous-tendu par un S.A.C. » qu'il s'agit donc d'identifier pour rendre compte (de la régulation) du fonctionnement de l'organisation.

1.2 – Le comportement des acteurs sociaux

La formalisation de la SAO a conduit à l'élaboration d'un méta-modèle de la structure des systèmes d'action concrets et au développement d'un environnement de simulation qui implémente ce modèle : *SocLab*.

La structure d'un SAC est représentée par un ensemble d'objets mathématiques bien définis, dont l'implémentation informatique dans *SocLab* permet à l'analyste d'une organisation d'en modéliser la structure sous la forme d'un système multi-agent, d'explorer son espace d'état (c'est-à-dire l'ensemble de tous les comportements que les acteurs pourraient adopter), d'étudier analytiquement les propriétés de cette structure, et de calculer un ensemble d'indicateurs qui font sens pour le sociologue, tels que la pertinence d'une ressource ou le pouvoir potentiel d'un acteur.

Puisque les jeux de pouvoir entre acteurs sont ce qui détermine la structure de leurs relations, la formalisation de la SAO conduit au développement d'un modèle de simulation de la rationalité des acteurs sociaux, dont les résultats indiquent comment il est plausible que les acteurs d'une organisation se comportent les uns vis-à-vis des autres. La sociologie de l'action organisée prête aux acteurs un comportement stratégique, c'est à dire objectivement finalisé vers un certain but, mais elle ne permet évidemment pas de prévoir le détail des moyens concret qu'un acteur utilisera pour réaliser ses objectifs. L'infinie diversité des situations concrètes et l'inventivité sans limite qu'une forte motivation peut susciter chez un acteur rendent définitivement imprévisibles les actes qu'il pourra entreprendre. Dire que les acteurs ont un comportement stratégique et choisissent rationnellement les actions qu'ils réalisent, c'est dire que ces actions sont sélectionnées sur la base de leur effet, de l'impact qu'elles auront sur le système. C'est donc l'effet des actions et non pas la forme particulière de leur réalisation qui est pertinent pour l'analyse sociologique et qu'il s'agit de modéliser. Le formalisme de modélisation permet de distinguer et de bien identifier, dans le modèle d'une organisation, ce qui relève de sa *structure* (ses éléments constitutifs, leurs relations et les opérations auxquelles ils peuvent donner lieu) et ce qui relève de son *état* dont l'ajustement synchronique accompagne la réalisation des finalités du système [Sibertin-Blanc *et al.*, 2004 , 2013]. Cela nous amène à distinguer deux dimensions dans le comportement d'un acteur : une dimension *structurelle* qui contribue à la construction d'un SAC et à faire évoluer sa structure, et une dimension *fonctionnelle* qui assure le fonctionnement régulier d'un SAC et fait évoluer son état. Ces deux dimensions sont indissociables l'une de l'autre dans l'action concrète d'un acteur ; chaque acte comporte une composante structurelle et une composante fonctionnelle, dans une proportion qui est spécifique aux conditions dans lesquelles cet acte est réalisé. La *dimension structurelle* de l'action est la part qui contribue à la construction du SAC comme organisation, à l'établissement des règles du jeu social et qui donc consiste, selon notre formalisation d'un SAC, à faire évoluer les acteurs, les relations, les contraintes et les enjeux qui constituent sa structure. Quant à la *dimension fonctionnelle* de l'action d'un acteur, c'est elle qui assure le fonctionnement régulier du système et fait évoluer son état ; elle concourt à la réalisation des objectifs de l'acteur. Cette dimension fonctionnelle s'exerce donc dans le contexte d'action défini par la structure de l'organisation, sans envisager de modification des règles du jeu ni de changement dans les objectifs (i. e. les enjeux) ou dans les moyens d'actions (i. e. les relations, leurs effets et leurs contraintes).

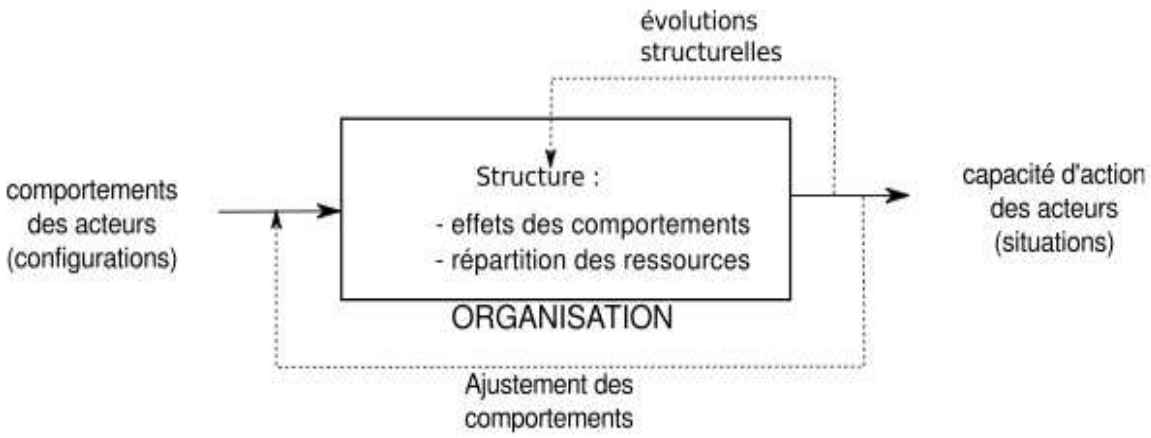


Figure 1.1. *L'organisation vue comme un système qui produit une situation régulée, en fonction de sa structure et des comportements possibles des acteurs ([Chapron, 2012], p. 12).*

La distinction entre ces deux dimensions de l'action n'a pas pour conséquence que l'on puisse envisager un SAC dans lequel les règles du jeu seraient figées, ou bien dans lequel ces règles du jeu seraient en évolution indépendamment des interactions directes entre les acteurs. L'identification de ces deux dimensions n'empêche pas de considérer qu'elles sont indissolublement liées, de façon dialogique, et que c'est dans cet espace à deux dimensions que l'action concrète des acteurs se déploie, cf. Figure 1.1. C'est dans l'atomicité de réalisation d'un acte concret que ces deux dimensions sont indissociables, mais l'effet d'un acte lui se décompose sans difficulté en une composante structurelle et une composante fonctionnelle. Puisque nous ne cherchons pas à modéliser ces modalités pratiques et ne nous intéressons qu'aux effets des actes, et que ces effets sur les deux dimensions structurelle et fonctionnelle sont bien disjoints, nous avons la possibilité de modéliser l'action des acteurs par des mécanismes spécifiques à chacune de ces dimensions. Le comportement « stratégique » d'un acteur vise d'une part à réaliser ses objectifs – c'est la dimension fonctionnelle – et d'autre part à conforter son pouvoir – c'est la dimension structurelle de son action.

La dimension structurelle du comportement des acteurs sociaux a fait l'objet de la thèse de Paul Chapron [Chapron, 2012], nous traitons dans cette thèse la dimension fonctionnelle.

Les algorithmes de simulation implémentés dans SocLab mettent en œuvre les principes de l'auto-apprentissage par essais-erreurs et le renforcement des règles apprises [Sutton *et al.*, 1998]. Ils se conforment au modèle d'acteur de la SAO, à savoir des acteurs stratégiques, disposant d'une rationalité limitée, et donc tâtonnant dans la recherche d'un comportement individuel qui soit raisonnablement satisfaisant. On attend donc des résultats de simulations qu'ils indiquent comment il est plausible que les acteurs d'une organisation se comportent les uns vis-à-vis des autres.

1.3 – Modèles de la rationalité des acteurs sociaux

Des versions successives d'un algorithme de simulation du comportement des acteurs sociaux ont été proposées et implémentées dans SocLab [Sibertin-Blanc *et al.*, 2005 ; Sibertin-Blanc *et al.*, 2006 ; Mailliard, 2008 ; El Gemayel *et al.*, 2011].

Dans cette thèse, nous présentons une version plus aboutie de cet algorithme et nous étudions sa validation. Notamment, cet algorithme comporte un ensemble de paramètres (taux de persévérance de l'acteur, capacité à discriminer les situations, etc.) dont le rôle et l'impact nécessitent d'être bien définis.

Tout d'abord, j'ai analysé l'effet de ces paramètres sur le processus d'apprentissage afin de ne garder que les plus significatifs, ce qui a conduit à la suppression de certains paramètres et à la

reformulation du rôle de certains autres. Ensuite, j'ai intégré de nouveaux paramètres qui semblent être intéressants du point de vue social. Et finalement, j'ai clarifié les paramètres d'apprentissage et psycho-cognitifs des acteurs sociaux et l'interdépendance entre ces paramètres.

S'agissant de simulation, qui plus est dans le domaine du sociale, la validation d'un tel algorithme est fondée sur les réponses aux questions suivantes :

- Est-il bien structuré et documenté, de telle sorte que l'on puisse comprendre son fonctionnement ?
- Est-il compatible avec ce que disent les sciences humaines et sociales ?
- Quelle confiance peut-on avoir dans les résultats qu'il fournit ?

En réponse aux deux premiers points, nous présentons cet algorithme de façon détaillé. A défaut d'établir qu'il est réaliste, nous montrerons qu'il est tout à fait vraisemblable, tant du point de vue social que du point de vue psycho-cognitif.

S'agissant de la qualité des résultats, on ne peut les comparer avec ce qui est attendu, avec ce qu'ils devraient être, lorsque les simulations portent sur une organisation mal connue, puisque l'objectif de ces simulations est justement d'apporter de nouvelles connaissances. Pour assurer la fiabilité de ces connaissances, il faut donc valider empiriquement l'algorithme en l'appliquant à des cas bien connus et bien compris pour lesquels on peut comparer les résultats de simulation avec ce qui est attendu. C'est ce que nous faisons en examinant soigneusement ce qu'il en est pour une diversité de modèles d'organisations qui reflètent, au moins partiellement, la diversité des organisations sociales.

Le modèle de rationalité des acteurs sociaux implanté dans SocLab est orienté vers la coopération : le jeu des interactions entre les acteurs se stabilise vers un optimum de Pareto (personne ne peut gagner plus sans que ce soit au détriment d'un autre). Cette thèse propose aussi d'autres modèles de rationalité des acteurs qui restent cognitivement vraisemblables et permettraient à une organisation de se stabiliser vers des formes de régulation correspondant à des archétypes familiers. Plusieurs pistes ont été étudiées et sont présentées dans les chapitres quatre et cinq, dont :

- Nash : chaque acteur joue défensif de façon à s'assurer d'obtenir le maximum qu'il puisse, quel que soit le comportement des autres acteurs. L'organisation se stabilise en un équilibre de Nash.
- Optimum Global : chaque acteur collabore avec les autres acteurs afin de maximiser la satisfaction globale de l'organisation : la somme des satisfactions de tous les acteurs.
- Elitiste (respectivement anti-élitiste) : chaque acteur joue de façon à augmenter (respectivement diminuer) la satisfaction de l'acteur le plus satisfait.
- Protecteur (respectivement anti-protecteur) : chaque acteur joue de façon à augmenter (respectivement diminuer) la satisfaction de l'acteur le moins satisfait.
- Egalitariste (respectivement anti-égalitariste) : chaque acteur joue de façon à réduire (respectivement agrandir) l'écart entre l'acteur le plus satisfait et l'acteur le moins satisfait.
- objectif multicritère : inspirée de [Selten, 1998], l'ambition d'un acteur est représentée comme un vecteur, afin de prendre en compte l'effet spécifique de chaque ressources et de ne pas agréger des aspects incommensurables : l'acteur cherche à améliorer a disponibilité de chacune des ressources dont il a besoin, au lieu de les agréger dans une grandeur globale, la satisfaction.

1.4 – L'organisation du document

Dans cette section, nous résumons les différents chapitres de cette thèse.

Dans le chapitre deux nous présentons le méta-modèle de la structure des organisations sociales, issu de la formalisation de la Sociologie de l'Action Organisée, qui détermine le contexte dans lequel les acteurs interagissent. Dans une première partie, nous présentons le cœur de ce méta-modèle et ses deux éléments constitutifs : les *Acteurs*, entités actives tout à la fois autonomes et contraintes, qui interagissent par l'intermédiaire de *Relations* fondées sur les ressources de l'action collective. Chacun des acteurs contrôle au moins une relation et a besoin de ressources contrôlées par d'autres acteurs. Le comportement plus ou moins coopératif du contrôleur de chaque relation détermine l'impact de cette relation sur les acteurs qui en dépendent. Il en résulte pour chaque acteur une certaine *capacité d'action* qui détermine les moyens dont il dispose pour réaliser ses objectifs.

Dans une deuxième partie, nous exposons des extensions de ce méta-modèle. Elles portent d'une part sur des contraintes sur les relations qui empêchent un acteur de fixer l'état d'une relation au-delà d'une certaine limite ou de modifier cet état à tout instant, et d'autre part sur des contraintes entre les relations qui permettent de limiter l'état d'une relation en fonction de l'état d'une autre. Cette extension comporte aussi des solidarités entre les acteurs qui permettent de représenter des liens entre les acteurs, par exemple des liens de parenté ou d'affinité.

Dans la troisième et la quatrième partie, nous présentons des modèles d'organisations virtuelles et d'organisations réelles qui seront utilisés pour les analyses de simulations et de sensibilités dans les chapitres quatre et cinq. L'intérêt des organisations virtuelles est qu'elles se contentent d'implanter un schéma d'interaction suffisamment simple pour que l'on puisse spécifier précisément les résultats que l'algorithme devrait produire, et que l'on puisse suivre les étapes par lesquelles il procède pour y parvenir. Ce sont donc des organisations qui nous permettent de caractériser la nature des résultats qui doivent être produits par l'algorithme en fonction des propriétés de structurelle de l'organisation auquel il s'applique, et les conditions de cette production. Les modèles d'organisations concrètes sont beaucoup plus complexes, les caractéristiques des relations que contrôle et dont dépend chaque acteur étant très spécifiques. Ce que l'algorithme doit produire peut alors être spécifié par une analyse sociologique du jeu des acteurs, même si c'est de façon beaucoup moins formelle et donc précise que dans le cas des organisations virtuelles. L'analyse des résultats produits par l'algorithme sur ce type de modèle permet de vérifier sa robustesse, qu'il reproduit raisonnablement bien les caractéristiques principales des résultats attendus, malgré l'enchevêtrement des logiques de comportement propres à chacun des acteurs.

Nous présentons ensuite le jeu social, dans lequel tous les acteurs cherchent à réaliser le méta-objectif consistant à disposer des moyens de réaliser ses objectifs concrets, qui est ce que notre algorithme de simulation cherche à modéliser. Puis nous présentons le comportement des acteurs sociaux tel qu'il est analysé par la SAO.

Le chapitre trois fait l'objet d'un état de l'art sur l'apprentissage en général, et plus précisément sur l'apprentissage par renforcement dans le cas d'un seul agent isolé et dans le contexte multi-agent. Dans une première partie, nous évoquons différentes approches d'apprentissage dont l'application n'est pas compatible avec le cadre d'analyse de la SAO. Nous introduisons chaque approche brièvement, puis nous citons quelques algorithmes pionniers ou récents dans le domaine, et nous concluons par les raisons pour lesquelles ces approches ont été exclues.

Nous introduisons ensuite l'apprentissage par renforcement, l'approche qui a été finalement retenue, puis nous détaillons la pertinence de cette approche dans notre contexte, ainsi que les défis rencontrés dans son application. Nous terminons par la présentation du contexte de l'interaction

entre un acteur et son environnement, tout d'abord sous la forme d'un processus décisionnel de Markov (MDP), puis étendu aux environnements non markoviens.

Dans une troisième partie, nous présentons les différentes méthodes d'apprentissage par renforcement : les méthodes à base d'un modèle, les méthodes d'apprentissage du modèle, et les méthode d'apprentissage sans modèle. Pour chacune de ces méthodes, nous rappelons le schéma de l'algorithme d'apprentissage, les conditions d'application de cette méthode, ses avantages et ses inconvénients.

La quatrième partie est consacrée pour l'apprentissage par renforcement dans les systèmes multi-agent. Tout d'abord, nous présentons une définition des systèmes multi-agents sous la forme d'un jeu stochastique à n acteurs, puis nous détaillons la pertinence et les défis de l'apprentissage par renforcement dans les SMA. Ensuite, nous citons quelques algorithmes pionniers ou récents dans le domaine, et nous concluons par une comparaison entre les domaines d'application des SMA et notre contexte.

Cet état de l'art nous donne la possibilité, dans la cinquième section, de caractériser les spécificités du jeu social.

Enfin, la dernière partie est consacrée à l'examen d'un certain nombre de modèles de la rationalité limitée qui ont été proposés comme alternatives au modèle de la rationalité parfaite.

Le chapitre quatre est consacré à l'algorithme de modélisation du processus de régulation des organisations, par la simulation des comportements que les acteurs sont susceptibles d'adopter afin de stabiliser l'organisation dans un état soutenable dans lequel chacun accepte la situation dans laquelle il se trouve. La première partie expose les principes de notre modélisation du comportements des acteurs et montre en quoi elle est cognitivement et socialement vraisemblable. La régulation d'une organisation est assimilée à un jeu dans lequel chacun des acteurs est autonome, agit de façon indépendante des autres, et cherche à obtenir les moyens de réaliser ses objectifs ; en d'autres termes à obtenir une « bonne » satisfaction mais pas nécessairement l'optimale.

La deuxième partie présente l'algorithme de simulation, qui décrit la façon dont les acteurs parviennent à déterminer comment se comporter les uns vis à vis des autres. Tout d'abord, nous introduisons les paramètres psycho-cognitifs qui permettent d'individualiser la façon dont chaque acteur délibère, les principales variables de l'algorithme, et les relations entres les deux. Ensuite, nous présentons les différentes étapes de cet algorithme, et nous discutons l'implémentation de certaines étapes. Nous concluons ainsi par une caractérisation des résultats de l'algorithme. Cet algorithme met en œuvre un mécanisme de résignation selon lequel, dans une organisation, quasiment tous les acteurs doivent plus ou moins réviser à la baisse leurs ambitions initiales afin de trouver un état qui les satisfassent. En effet, l'ambition d'un acteur, qui représente son objectif, n'est pas prédéfinie, mais évolue contextuellement ; plus cet objectif est éloigné, plus l'acteur va explorer en réalisant des actions énergétiques, et moins il est enclin à réviser son ambition. Contrairement, si un acteur est satisfait, c'est-à-dire qu'il a atteint son objectif, il augmente son ambition : se rendant compte de ce qu'il est capable de faire, il révise son ambition à la hausse à proportion de ses capacités. Dés que tous les acteurs sont satisfaits, nous considérons que l'algorithme a convergé vers un état stationnaire qui est accepté par tous les acteurs ; l'organisation est alors régulée.

Dans la troisième partie, nous présentons quelques méthodes d'analyse statistique utilisées pour interpréter les résultats de simulation et des analyses de sensibilités : l'analyse univariée qui étudie la distribution de chaque variable indépendamment des autres, l'analyse multivariée qui permet d'étudier les corrélations entres les différentes variables, et les méthodes de classification. En effet, la dimension stochastique de l'algorithme conduit à réaliser plusieurs simulations pour chaque modèle afin de constituer un jeu de résultats significatif et d'éviter les cas particuliers. L'utilisation des outils de l'analyse statistiques est donc nécessaire.

La quatrième partie est consacrée à l'analyse des résultats de simulation produits par l'algorithme sur trois modèles d'organisations virtuelles et trois modèles d'organisations réelles, afin de vérifier que les résultats sont bien ceux attendus. Ensuite, dans la cinquième partie, nous faisons une analyse de sensibilité pour évaluer l'impact de chacun des paramètres psycho-cognitifs, afin de discuter leur influence sur les résultats en fonction de leur valeur et des caractéristiques de la structure de l'organisation.

La dernière partie est consacrée à une extension de cet algorithme qui permet à l'acteur de prendre en considération la poursuite de plusieurs objectifs incomparables. Nous exposons l'algorithme qui résulte de modifications apportées à l'algorithme général. Nous présentons ensuite les résultats produits par cet algorithme sur différents modèles, qui s'avèrent assez peu différents des résultats de l'algorithme initial.

Dans le chapitre cinq, nous présentons cinq modèles de rationalité, adaptations de l'algorithme présenté dans le chapitre quatre. Chacun de ces modèles de rationalité considère que le comportement de l'acteur n'est pas exclusivement guidé par la recherche de son intérêt personnel, *i.e.* la maximisation de sa satisfaction, mais qu'il inclut la recherche d'un certain ordre social. Lorsqu'il est adopté par certains ou tous les acteurs d'une organisation, chacun de ces algorithmes devrait conduire l'organisation à se réguler vers des configurations satisfaisant une propriété spécifique. Pour chaque type de rationalité, nous présentons un algorithme de la rationalité des acteurs qui est une variante de l'algorithme présenté dans le chapitre 4. Nous analysons en détail les résultats obtenus par cet algorithme sur une variété de modèles d'organisations virtuelles et réelles.

Enfin, **le chapitre six** est consacré à SocLab, l'environnement développé en java pour permettre de créer et d'éditer des modèles d'organisations, d'en analyser les états remarquables, d'e connaître les valeurs d'indicateurs structurels sur les acteurs, les relations et les liens entre eux, d'effecteur des simulations et des analyses de sensibilité, et enfin de produire différents supports de présentations de ces résultats (tableaux, réseaux, courbes, …) afin de faciliter leur interprétation.

Le chapitre sept est consacré à la conclusion et aux prolongements possibles des travaux présentés dans ce mémoire.

Chapitre 2 – Un Méta-modèle des systèmes d'actions concrets

Formaliser la théorie de la Sociologie de l'Action Organisée consiste à proposer un cadre formel pour modéliser et analyser les Systèmes d'Action Concret [Sibertin-Blanc *et al.*, 2005, 2013]. Dans ce chapitre, nous présentons le méta-modèle des systèmes d'action concrets. Il est proposé comme une formalisation de la SAO, entreprise en collaboration avec des sociologues [Roggero *et al*. 2008]. Chaque modèle d'une organisation particulière est donc une instance de ce méta-modèle.

Nous présentons tout d'abord le socle du méta-modèle (§2.1) ainsi que certaines de ses extensions (§2.2) en nous inspirant très largement de [Mailliard, 2008] et [Chapron, 2012]. Puis nous présentons trois exemples de modélisation d'organisations virtuelles (§2.3) et trois exemples de modélisation d'organisations réelles (§2.4). Ensuite nous présentons le jeu social (§2.5). Finalement nous présentons le comportement des acteurs sociaux tel qu'il est analysé par la SAO (§2.6) et les approches liées (§2.7).

2.1 – Le socle du méta-modèle des organisations

Si l'on s'en tient aux concepts structurants de la Sociologie de l'Action Organisée, trois éléments se dégagent : les Ressources ou zones d'incertitude, les Relations et les Acteurs. Cette sociologie est bien une sociologie d'acteurs entretenant des relations en manipulant des ressources leur conférant du pouvoir. Comme l'écrit Friedberg: « pas de pouvoir sans relation, pas de relation sans échange » ([Friedberg, 1993], page 115). Le pouvoir suppose donc la relation qui implique l'échange qui lui-même nécessite des objets d'échange : les ressources.

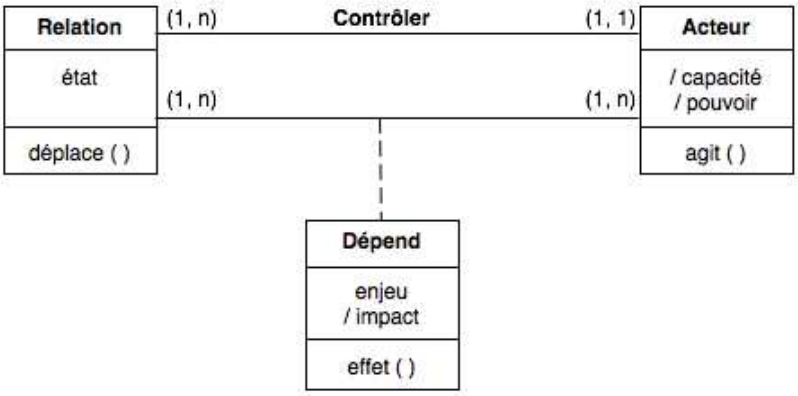


Figure 2.1. *Diagramme de classe UML du méta-modèle des SAC.*

Le diagramme de classes UML de la figure 2.1 est une représentation d'un méta-modèle simplifié des SAC, dont les éléments constitutifs sont les *Acteurs* et les *Relations*, reliés par les associations *Contrôler* et *Dépendre*. Plusieurs acteurs sont liés à chaque relation : celui qui la contrôle et ceux qui en dépendent (parmi lesquels généralement celui qui la contrôle). Chaque relation est *contrôlée* par un acteur unique qui est le seul à pouvoir modifier son *état*. Chaque acteur place un certain *enjeu* sur certaines relations, celles dont il *dépend*. Dans ce cas, il en obtient un certain impact, qui est déterminé par l'application de la *fonction d'effet* à l'état de la relation, pondéré par l'enjeu sur cette relation. Finalement, il en résulte pour l'acteur une certaine *capacité d'action*, agrégation des impacts qu'il reçoit sur les relations dont il dépend, et un certain *pouvoir*, agrégation des capacités d'action qu'il attribue aux acteurs qui dépendent des relations qu'il contrôle.

2.1.1 – Les ressources et leurs relations

Les Ressources d'un SAC sont, dans le sens le plus large du terme, les éléments dont la disponibilité est requise pour réaliser certaines actions dans le contexte d'action collective. Chaque acteur maîtrise ainsi au moins une ressource qui elle-même peut être maîtrisée par plusieurs acteurs. Une ressource fonde une ou plusieurs relations, et une relation est fondée par une seule ressource (figure 2.2).

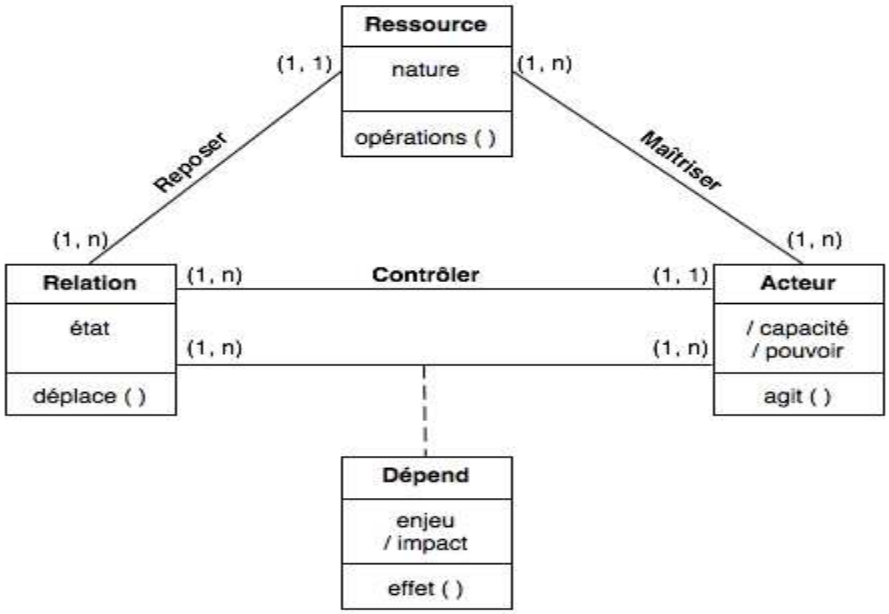


Figure 2.2. *Diagramme de classe UML du méta-modèle des SAC avec les ressources, fondements des relations.*

Au niveau du méta-modèle, les ressources sont avant tout des éléments dont le repérage permet de guider l'identification des relations et des acteurs. Elles permettent de fournir et de maintenir le lien sémantique existant entre les données de terrain et le modèle mais elles ne sont pas réifiées au sein de l'implémentation du modèle. En effet, la méthodologie que propose la SAO consiste à repérer les ressources pertinentes d'une organisation afin de trouver les acteurs pertinents et les relations qui les lient. Elles ne sont que des éléments du contexte du modèle d'une organisation et le méta-modèle peut s'exprimer comme uniquement constitué de relations et d'acteurs (figure 2.1).

Une Relation, donc, correspond à un certain type de transactions concernant la Ressource sur laquelle elle est fondée, et elle est déséquilibrée : un (unique) acteur - l'un de ceux qui maîtrisent la Ressource - contrôle cette relation, tandis que d'autres acteurs – ceux qui ont besoin de cette Ressource pour atteindre leurs objectifs – sont contrôlés, dominés, ou encore dépendants de cette Relation. En effet, c'est l'Acteur qui contrôle la Relation qui détermine par son comportement dans quelle mesure la Ressource est accessible aux autres. Il contrôle ainsi la possibilité pour les acteurs dépendants de réaliser leurs objectifs.

2.1.2 – L'état d'une relation et ses effets

Un SAC est constitué de relations entre des acteurs. Soit R l'ensemble des relations du SAC. L'*état* e_r d'une relation $r \in R$ est un réel défini dans l'espace d'état E_r, fixé arbitrairement comme étant une échelle bi-polaire sur l'intervalle [-10 ; 10]. L'état de chaque relation est modifiable par un et un seul acteur, le *contrôleur* de cette relation. L'état e_r d'une relation r représente le comportement que le contrôleur de la relation adopte pour celle-ci, et l'espace d'état E_r représente l'espace des comportements potentiels du contrôleur. L'interprétation de l'état d'une Relation est à définir en terme de comportement de l'Acteur qui la contrôle : les extrémités –10 et +10 de l'espace

des choix correspondent aux limites techniques, de faisabilité ou physiques tenant à la nature de la relation.

L'état d'une relation a un *effet* sur chacun des acteurs qui en dépend ; il lui procure une certaine *capacité d'action* (voir 2.1.4) pour atteindre ses objectifs : plus l'effet d'une relation sera élevé, plus ou mieux la relation sera utilisable par l'acteur. Les effets sont gradués sur une échelle de -10 à 10, elle aussi arbitraire :

Pire des cas = -10 neutre = 0 optimal = 10

Les effets d'une relation selon son état sont déterminés par une fonction d'effet qui est déterminée par la nature de cette relation et son utilité pour l'acteur :

$$effet_r : A \times E_r \rightarrow [-10 ; 10]$$

où A est l'ensemble des acteurs. L'unité de mesures des effets est la capacité d'action. Le choix (par l'acteur contrôleur) d'une valeur $e_r \in E_r$ dans l'espace des comportements de la relation r se traduit donc par un $effet_r (a, e_r)$ pour l'acteur a.

2.1.3 – Les acteurs et leurs enjeux

La sociologie de l'action organisée postule que chaque acteur, quelque soit sa position, a la possibilité d'exercer sa maîtrise sur certaines ressources, fussent-elles mineures. Tout acteur contrôle donc une ou plusieurs relations ; c'est reconnaître à tout acteur d'un système d'action concret une certaine marge de manœuvre, celle qu'il exerce sur les relations qu'il contrôle, et par là même dénier le statut d'acteur à toute personne qui ne contrôlerait aucune relation. Les notions de relation et d'acteur se définissent l'une par rapport à l'autre : une relation n'est telle que dans le mesure où certains acteurs en dépendent pour réaliser leur activité ; est acteur celui qui contrôle une relation.

Il en résulte que, dans la modélisation d'un système d'action concret, la correspondance entre acteur et personne physique n'a rien d'automatique. Si un individu qui ne contrôle aucune Relation n'est pas un Acteur, une population homogène peut être agrégée en un Acteur unique si chacun de ses membres est dans la même situation de dépendance et de contrôle vis-à-vis de toutes les relations (et ce avec les mêmes enjeux), si bien qu'ils auront des comportements similaires. C'est l'analyse sociologique qui identifie les acteurs du système d'action concret, à un niveau de granularité qui dépend de la finalité de cette analyse.

D'autre part, la SAO considère que le comportement des acteurs sociaux est *stratégique*. Considérer un acteur comme stratégique revient à lui attribuer un comportement intéressé, c'est-à-dire « motivé par une visée, sans préciser davantage la nature de cette visée ou de ce mobile » ([Friedberg, 1993], page 214-215). Cette « visée », ou objectif, l'amène à mobiliser les moyens dont il dispose pour tenter d'atteindre cette dernière et donc à jouer des relations dans lesquelles il est impliqué. Chacune de ces relations aura plus ou moins de valeur à ses yeux, c'est ce que traduit la notion d'*enjeu*.

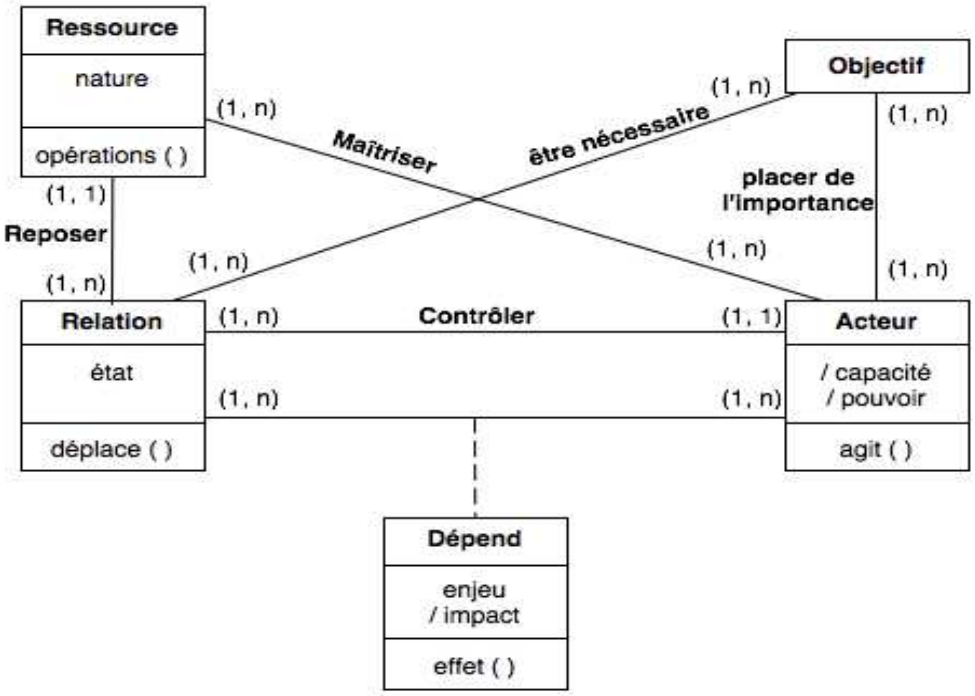


Figure 2.3. *Diagramme de classe UML du méta-modèle des SAC avec les objectifs des acteurs.*

Chaque Acteur répartit son capital d'enjeux sur les relations, en fonction de l'importance que revêt chaque Relation pour atteindre ses objectifs: plus l'usage de la Ressource accessible via la Relation est nécessaire pour atteindre un objectif qui est important pour l'Acteur, plus l'enjeu que l'Acteur place sur cette Relation est élevé. Cette répartition des enjeux d'un Acteur correspond à l'impact opérationnel de ses objectifs sur son comportement. Pour l'analyse du fonctionnement d'un système d'action concret, ce qui importe ce n'est pas tant la nature des objectifs d'un acteur que ce qu'ils le conduisent à faire. Les enjeux sont le maillon qui, conformément à l'hypothèse de rationalité des acteurs, permet de relier causalement le comportement d'un acteur avec ses objectifs, comme le montre la figure 2.3.

La distribution des enjeux sur une échelle de valeurs numériques permettra de dégager des indicateurs synthétiques quantitatifs. Les enjeux sont des coefficients sur la capacité d'action, gradués sur une échelle de 0 à 10, là encore arbitraire :

Nul = 0 Important = 5 Vital = 10

Nous attribuons à chaque acteur la même quantité de points d'enjeux à répartir, fixée arbitrairement à 10, selon l'idée que tous les acteurs d'une organisation ont le même investissement dans le jeu social, quelque soit leur position.

Suivant la valeur de l'enjeu d'un acteur pour une relation et la valeur de l'état de la relation, il est alors possible de déterminer l'*impact* que la relation aura sur la capacité de l'acteur à atteindre ses objectifs :

$$\text{impact}(a, r, e_r) = \text{enjeu}(a, r) \times \text{effet}_r(a, e_r)$$

2.1.4 – La capacité d'action et le pouvoir d'un acteur

Ce que vise un Acteur, c'est de disposer des moyens nécessaires à la réalisation de ses objectifs[1]. Une grandeur particulièrement significative pour un Acteur est alors la capacité d'action dont il dispose pour réaliser ses objectifs, évaluée comme l'agrégation des impacts des relations dont il dépend. Nous l'appellerons la *capacité d'action* d'un Acteur (de préférence au terme d'utilité couramment employé en économie, en ce qu'il est plus descriptif et évocateur d'une rationalité limitée). Elle reflète la possibilité qu'a un Acteur d'accéder aux relations dont il a besoin pour atteindre ses objectifs. De ce fait, elle mesure le niveau de réalisation d'une sorte de méta-objectif commun à tous les acteurs sociaux : obtenir les moyens de ses objectifs. Cette capacité d'action peut être définie comme la somme des impacts des relations dont un acteur dépend :

$$capacitéAction(a,e) = \sum_{r \in R} impact(a,r,e_r) = \sum_{r \in R} enjeu(a,r) \times effet_r(a,e_r)$$

Une autre grandeur très significative à considérer est la mesure dans laquelle un Acteur contribue à la capacité d'action d'un autre, c'est-à-dire la quantité de capacité d'action qu'il lui prodigue. C'est ce qui semble le mieux exprimer la notion de pouvoir qui est au cœur de la sociologie de l'action organisée. Nous pouvons alors quantifier le pouvoir qu'un Acteur « a » exerce sur les autres dans un état « e » du système d'action de la façon suivante :

$$pouvoir(a,e) = \sum_{r \in R \,/\, contrôle(r)=a} \sum_{b \in A} impact(b,r,e) = \sum_{r \in R \,/\, contrôle(r)=a} \sum_{b \in A} enjeu(b,r) \times effet_r(b,e)$$

2.1.5 – Représentation mathématique de la structure d'un SAC

A partir de ce qui précède, nous pouvons formaliser la définition d'un système d'action concret sous forme mathématique, de la façon suivante :

Terminologie

- $A = \{a_1, ..., a_N\}$, l'ensemble des N acteurs du jeu.
- $R = \{r_1, ..., r_M\}$, l'ensemble des M relations du jeu.
- $S = [-10 ; 10]^M$, est l'ensemble des états du jeu. Un état est représenté comme un vecteur $(e_r)_r$ constitué de l'état er de chaque relation r constitutive du système, $r \in R$. Rappelons que l'état e_r d'une relation est définit sur [-10 ; 10].

Fonctions

- enjeu : A x R → [0, 10], une fonction qui indique l'enjeu que chaque acteur place sur chacune des relations, avec $\forall a \in A, \Sigma_{r \in R}$ enjeu (a, r) = 10.
- contrôle : R → A, une fonction qui indique quel est l'Acteur qui contrôle chacune des relations.
- effet : R x A x [-10, 10] → [-10, 10] : une fonction qui retourne l'effet d'une relation r sur un acteur a pour un état e donnée entre -10 et 10.
- Transition : Etat x Action → Etat

  $(e_{r1}, ... , e_{rM})$ x $(d_{r1}, ..., d_{rM})$:→ $(e_{r1} + d_{r1}, ..., e_{rM} + d_{rM})$

  où Action est l'ensemble des actions que peuvent réaliser les acteurs sur les relations qu'ils contrôlent, d_{ri} étant fixé par l'Acteur qui contrôle la relation r_j.

[1] La mise en cohérence des fins et des moyens peut également consister à modifier ses objectifs en fonction des ressources mobilisables.

Un S.A.C. apparaît en fait comme étant un automate, dont les acteurs peuvent modifier l'état en agissant sur l'état des relations qu'ils contrôlent. Les actions que peut réaliser un Acteur consistent à déplacer l'état des relations qu'il contrôle tout en respectant les limites (bornes inférieures et supérieures) de chacune des relations.

2.2 – Une extension du méta-modèle des organisations

L'utilisation de ce méta-modèle pour modéliser des SAC, par exemple les cas d'école présentés dans [Bernoux, 1985], nous a conduit à le compléter avec différents éléments que nous présentons dans cette section (figure 2.4).

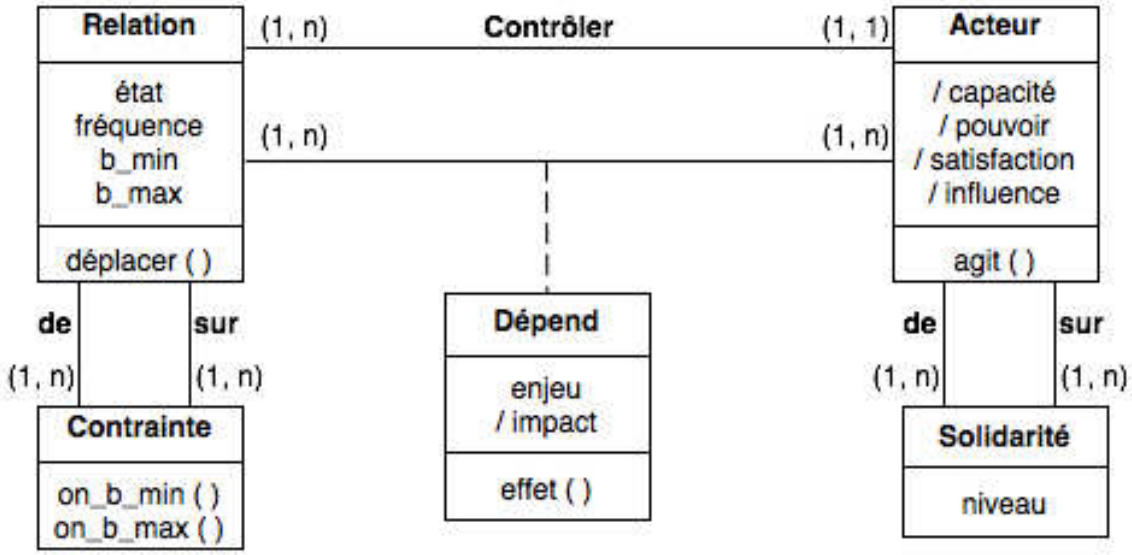


Figure 2.4. *Diagramme de classe UML du méta-modèle complet des SAC.*

2.2.1 – Les contraintes sur l'état d'une relation

L'acteur qui contrôle une relation ne peut pas pour autant attribuer n'importe quels effets aux acteurs dépendant de cette relation. Il doit respecter « les règles » du jeu social qui déterminent, pour partie, son comportement et donc l'état des relations qu'il contrôle.

C'est pourquoi, à l'intérieur de l'espace de comportement de chaque relation, il existe d'autres contraintes dont l'origine est soit institutionnelle par les règles formelles internes ou imposées à l'organisation, soit encore tenant à l'acceptabilité sociale en fonction des normes en vigueur [Kellerhals *et al.*, 1988]. Nous proposons de formaliser ces contraintes sur l'état d'une relation, d'une part par deux variables *b_min* et *b_max* telles que $-10 \leq b\_min \leq b\_max \leq 10$, et d'autre part par une variable *fréquence* qui correspond au fait qu'un acteur n'est pas en mesure d'accéder à la relation à tout instant pour modifier son état. Une valeur entre 0 et 1 est attribuée à la fréquence de chaque relation indiquant la probabilité d'accès à cette relation par le contrôleur.

Les attributs *b_min* et *b_max* d'une Relation décrivent les contraintes auxquelles est soumis l'Acteur qui contrôle cette relation. Nous dirons que l'intervalle [*b_min* ; *b_max*] est la marge de manœuvre de l'acteur qui contrôle la relation et son amplitude (*b_max* - *b_min*) représente l'*étendue* de son contrôle sur la relation.

2.2.2 – Les contraintes entre les relations

Dans le SAC paradigmatique de la SAO, à savoir « le monopole industriel » en l'occurrence la SEITA ([Crozier 1963] pp. 67-174 et 186-214), il y a une relation qui porte sur « l'entretien des machines », maîtrisée par les « ouvriers d'entretien », et une autre relation qui porte sur « la

production », maîtrisée par les « ouvriers de production ». Il s'avère impossible de modéliser correctement ce cas en ignorant que la façon dont les ouvriers d'entretien s'occupent des machines détermine la marge de manœuvre des ouvriers de production : si les machines fonctionnent mal, les ouvriers de production peuvent difficilement s'investir dans la production, quelle que soit leur bonne volonté.

Cela conduit à introduire les contraintes qu'une relation peut exercer sur une autre en agissant sur son intervalle [b_min, b_max] et donc l'étendue du contrôle de cette dernière. La contrainte qu'une relation r exerce sur une relation r' peut s'exprimer par l'intermédiaire de deux fonctions :

$$on\_b\_min_{r, r'} : [-10 ; 10] \rightarrow [-10 ; 10] \text{ et } on\_b\_max_{r, r'} : [-10 ; 10] \rightarrow [-10 ; 10]$$

telles que dans l'état $e = (e_{r1}, \ldots, e_{rn})$ de l'organisation :

$$b\_min_{r'} = \max_{r \,/\, r \text{ contraint } r'} \{on\_b\_min_{r, r'} (e_r)\}$$

$$\text{et } b\_max_{r'} = \min_{r \,/\, r \text{ contraint } r'} \{on\_b\_max_{r, r'} (e_r)\}$$

2.2.3 – Les solidarités

Les ressources d'une organisation, et la façon dont elles sont instrumentées en relations par les acteurs, ne suffisent pas toujours à rendre compte du comportement des acteurs. Il peut en effet exister entre les acteurs des liens familiaux (par exemple dans le cas « Bolet » [Bernoux, 1985]), des solidarités de condition sociale (de classe ou d'éducation, etc.) qui sont extérieures à l'organisation, ainsi que des convergences/divergences d'intérêts internes explicitement reconnues par les acteurs.

Cela conduit à introduire les *solidarités* que les acteurs peuvent avoir les uns envers les autres sous la forme d'une fonction :

$$\text{solidarité } (a, b) : A \times A \rightarrow [-1, 1] \text{ telle que } \forall\, a \in A, \text{ solidarité } (a, a) = 1$$ [2]

où les valeur négatives représentent des inimitiés, nulles l'indifférence et positives les solidarités.

Ces coefficients permettent de quantifier dans quelle mesure un acteur prend en compte non seulement sa propre capacité mais aussi celle des acteurs dont il est solidaire positivement (coefficient compris entre 0 et 1) ou négativement (coefficient entre -1 et 0).

La prise en compte des solidarités conduit à introduire, pour caractériser la situation d'un acteur dans un état de l'organisation, deux autres grandeurs en complément de la capacité et du pouvoir introduits en 2.1.4 :

$$Satisfaction(a,e) = \sum_{b \in A} solidarité(a,b) \times capacitéAction(b,e)$$

$$Influence(a,e) = \sum_{c \in A} \sum_{b \in A} solidarité(c,b) \times \sum_{r \in R \,/\, contrôle(r)=a} impact(b,r,e_r)$$

Satisfaction et influence cherchent à rendre compte de la façon dont chaque acteur se représente le jeu social en fonction de ses liens avec les autres acteurs, alors que capacité et pouvoir expriment ce qu'il en est effectivement de la situation de chacun. En l'absence de solidarités, lorsque chaque acteur est solidaire exclusivement de lui même, la satisfaction se ramène à la capacité et l'influence au pouvoir.

---

[2] On peut aussi normaliser la solidarité des acteurs pour rendre leurs satisfactions comparables en imposant la contrainte $\sum_{b \in A} solidarité(a, b) = 1$ ou bien $\sum_{b \in A} |solidarité(a, b)| = 1$; discuter les interprétations correspondantes nécessiterait un long développement.

2.2.4 – Le contrôle partagé des relations

Puisque c'est chaque acteur, in fine, qui décide de son propre comportement, il ne saurait être question, selon la SAO, qu'une relation soit contrôlée par plusieurs acteurs. Rien n'empêche cependant de considérer une telle situation. Il suffit pour cela de déterminer comment le contrôle d'une relation se répartit entre les acteurs, par une fonction :

$$\text{Contrôle}(r, a) : R \times A \rightarrow [0, 1] \text{ telle que } \forall r \in R, \Sigma_{a \in A} \text{ contrôle}(r, a) = 1$$

Dans ce cas, si $d_{ai,r}$ est l'action de l'acteur a_i sur la relation r, le nouvel état e'_r de r résultant de l'exécution de cette action est déterminé selon le pourcentage de contrôle partagé :

$$e'_{r_j} = e_{r\,j} + \sum_{a_i \in A} contrôle(r, a_i) \times d_{a_i, r}$$

2.3 – Des exemples de modèles d'organisations virtuelles

Dans cette section, nous présentons une version sociale du très classique dilemme du prisonnier, ainsi qu'une extension à n acteurs. Ensuite, nous présentons un modèle free-rider.

2.3.1 – Le dilemme de prisonnier social classique

Le Dilemme du Prisonnier (DP) est un jeu proposé dans les années 50 par les mathématiciens Merill Flood et Melvin Dresher. Il se présente comme un jeu symétrique à deux joueurs (cf. tableau 2.1), ici des prisonniers, ayant chacun la possibilité de jouer soit la coopération (C) soit la Défection (D). Comme montré en table 2.1, chaque joueur reçoit une rétribution qui dépend du choix de chacun des deux joueurs. Si les deux joueurs coopèrent, ils recevront la récompense pour avoir coopéré (R) ; si les deux jouent la trahison, ils en seront punis (P) ; et si l'un coopère tandis que l'autre trahit, il est le stupide (S) tandis que l'autre recevra la rétribution de sa Trahison (T).

	C	D
C	(R, R)	(S, T)
D	(T, S)	(P, P)

Tableau 2.1. *La matrice des rétributions pour le jeu du dilemme du prisonnier.*

Le DP est un dilemme lorsque la tentation est plus profitable que la coopération mutuelle, qui rapporte plus que la punition, qui est plus avantageux que d'être le joueur stupide : $T > R > P > S$, et que de plus la somme des gains pour la coopération mutuelle (C, C) est plus importante que celle pour la trahison/coopération (D, C) : $R + R > S + T > P + P$. Ainsi, le dilemme est que la stratégie individuellement rationnelle (celle qui minimise les pertes) est de jouer la trahison, ce qui conduit les deux joueurs à l'équilibre de Nash (D, D), alors que la meilleure stratégie collective (celle qui maximise le total des gains) est de jouer la coopération mutuelle (C, C), qui conduit à l'optimum de Pareto.

Dans la version itérée du DP, chaque joueur peut appliquer des choix différents dans les confrontations successives et les rétributions sont sommées. La version itérée du DP a été largement explorée et exposée ([Hoffman, 2000], [Delahaye, 1992], [Dugatkin, 1997], [Macy *et al.*, 2002] parmi beaucoup d'autres) depuis le fameux tournoi proposé par Axelrod [Axelrod, 1992].

En ce qui concerne notre dilemme du prisonnier social, considérons un SAC comportant deux acteurs, A1 et A2, et deux relations R1 et R2 tels que A1 contrôle R1 et A2 contrôle R2. Par contre, A1 place davantage d'enjeux sur R2 que sur R1, et A2 sur R1 que sur R2, comme par exemple dans le tableau 2.2. De plus, les fonctions d'effets des relations sont linéaires et symétriques, avec une

pente -1 sur la relation que l'acteur contrôle et 1 sur l'autre relation, comme indiqué dans le tableau 2.2.

Enjeux	A1	A2
R1	**1**	9
R2	9	**1**

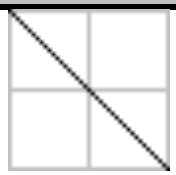
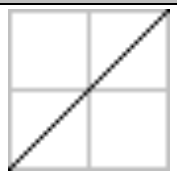
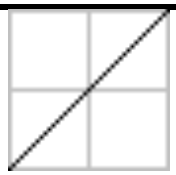
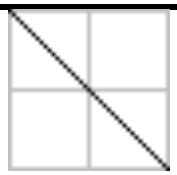
Effets	A1	A2
R1		
R2		

Tableau 2.2. *Les enjeux des acteurs sur les relations dont ils dépendent (les contours renforcés indiquent le contrôleur de chaque relation), et les fonctions d'effet des relations sur les acteurs (l'axe des x est l'espace d'états de la relation, et l'axe des y est l'impact de la relation sur l'acteur).*

Le tableau 2.3 montre les satisfactions respectives des deux acteurs pour les valeurs typiques des états des relations. Si l'on considère la satisfaction globale, qui est la somme des satisfactions des deux acteur, le meilleur cas, c'est quand les deux acteurs coopèrent – ils abandonnent la satisfaction qu'ils pourraient obtenir de la relation qu'il contrôle pour satisfaire l'autre –, alors que le pire des cas, qui est l'équilibre de Nash, c'est quand ils ne coopèrent pas. Quand un acteur coopère et l'autre trahit, ce dernier obtient 100 tandis que celui qui a coopéré obtient -100. Toute répartition des enjeux dans laquelle davantage d'enjeu est placé sur la relation contrôlée par l'autre acteur constitue un dilemme du prisonnier. Bien que ce modèle d'organisation soit théorique, il remplit une propriété essentielle des jeux sociaux réels : ils ne sont pas des jeux à somme nulle, et la meilleure satisfaction globale, d'une part, nécessite une coopération des acteurs, et d'autre part, est compatible avec une satisfaction importante des deux acteurs [Clegg, 2003].

		Etat de R1		
		-1	**0**	**+1**
Etat de R2	**-1**	-80 / -80	-90 / 10	-100 / 100
	0	10 / -90	0 / 0	-10 / 90
	+1	100 / -100	90 / -10	80 / 80

Tableau 2.3. *Satisfaction (A1) / satisfaction (A2) dans les états caractéristiques du dilemme du prisonnier social.*

2.3.2 – Le dilemme de prisonnier social à n acteurs

Nous présentons ici une extension du dilemme du prisonnier à n acteurs où chacun dépend du suivant. Considérons un SAC comportant n acteurs et n relations tels que :

- A1 contrôle R1, … An contrôle Rn.
- A1 place davantage d'enjeux sur R2 que sur R1, …, An-1 place davantage d'enjeux sur Rn que sur Rn-1, et An place davantage d'enjeux sur R1 que sur Rn.
- Les fonctions d'effets des relations sont linéaires et symétriques, avec une pente -1 sur la relation que l'acteur contrôle, une pente 1 sur l'autre relation dont il dépend, et une pente 0 sur les autres relations.

Il s'agit d'un dilemme de prisonnier circulaire, où A1 dépend de A2, …, An-1 dépend de An, et An dépend de A1, comme présenté dans la figure 2.5.

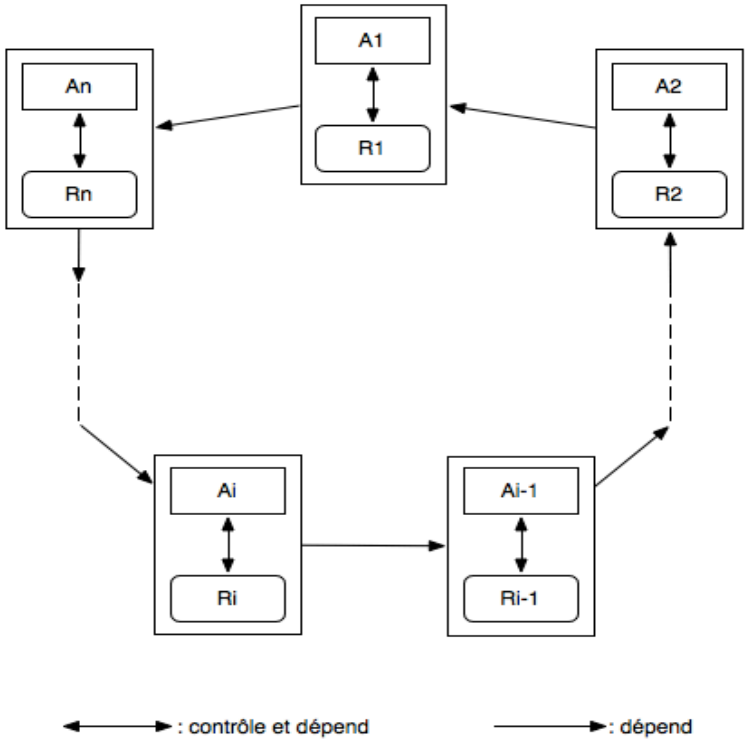


Figure 2.5. *La structure d'un dilemme du prisonnier circulaire où n est le nombre des acteurs et des relations.*

Comme dans le dilemme du prisonnier social classique, la meilleure configuration, qui maximise la satisfaction globale, est celle dans laquelle tous les acteurs coopèrent, alors que le pire des cas, qui est l'équilibre de Nash, est la configuration contraire dans laquelle aucun acteur ne coopère. Par contre, ce jeu présente une difficulté de coordination qui s'agrandit avec le nombre des acteurs : quand An coopère avec An-1, pour que An ne soit pas le stupide et donc découragé de coopérer, il faut que An-1 coopère avec An-2, et ainsi de suite jusqu'à ce que A1 coopère aussi avec An. Il faut donc que tous les acteurs adoptent en même temps un comportement coopératif pour que l'organisation converge vers un état d'équilibre qui satisfait tous les acteurs.

2.3.3 – Un modèle « free-rider »

Ce modèle comporte quatre acteurs et quatre relations et il conduit assez naturellement l'un des acteurs de faire défection tout en profitant de la coopération des autres. Chacun des quatre acteurs contrôle une seule relation ; l'acteur A1 contrôle la relation R1 et dépend des trois autres relations (R2, R3, R4) contrôlées respectivement par les acteurs A2, A3, et A4, qui, pour leur part, dépendent tous de la relation R1. L'acteur A1 est donc globalement confronté aux trois acteurs A2, A3, et A4, qui sont indépendants les uns des autres. La figure 2.6 indique les contrôles, les enjeux et les fonctions d'effet de ce jeu.

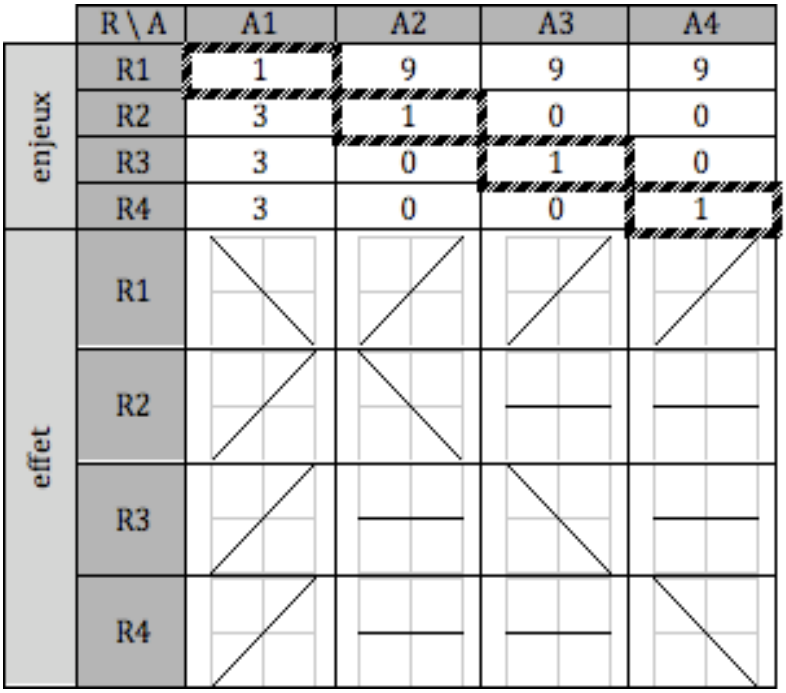


Figure 2.6. *Les enjeux (les cases bordées de noir indiquent quel acteur contrôle la relation) et les fonctions d'effet (l'axe des x correspond à l'espace de comportement de la relation, l'axe des y à l'effet sur l'acteur) du modèle free-rider.*

Le tableau 2.4 indique les satisfactions des acteurs pour des configurations remarquables du système, dont les huit premières configurations correspondent à des optimums de Pareto, tandis que la dernière configuration (C9) correspond à un équilibre de Nash qui se trouve minimiser la satisfaction globale. Le maximum de la satisfaction globale (C1) est atteint par la coopération de chacun des acteurs, et ce au détriment du maximum de ce à quoi chacun pourrait prétendre (C8, C2, C3, C4 respectivement). Remarquons que le maximum de A1 (C8) nécessite la coopération des trois autres acteurs et les pénalise très fortement. Les configurations C5, C6 et C7 correspondent à un état qui maximise la satisfaction de deux acteurs parmi A2, A3 et A4, et pénalise fortement A1.

Configurations		C1	C2	C3	C4	C5	C6	C7	C8	C9
Etat des relations	R1	1	1	1	1	1	1	1	-1	-1
	R2	1	-1	1	1	-1	-1	1	1	-1
	R3	1	1	-1	1	-1	1	-1	1	-1
	R4	1	1	1	-1	1	-1	-1	1	-1
Satisfaction des acteurs	A1	80	20	20	20	-40	-40	-40	**100**	-80
	A2	80	**100**	80	80	100	100	80	-100	-80
	A3	80	80	**100**	80	100	80	100	-100	-80
	A4	80	80	80	**100**	80	100	100	-100	-80
	Somme	**320**	280	280	280	240	240	240	-200	-320

Tableau 2.4. *Satisfactions correspondants à neuf états particuliers du système : l'état qui maximise la satisfaction globale (C1) ; les quatre états qui maximisent la satisfaction d'un des quatre acteurs : A1 (C8), A2 (C2), A3 (C3) et A4 (C4) ; trois états dans lesquels deux des trois acteurs ne coopèrent pas : A2 et A3 (C5), A2 et A4 (C6), A3 et A4 (C7) ; l'équilibre de Nash (C9).*

2.4 – Des exemples de modélisation d'organisations réelles

Dans cette section, nous présentons trois modèles d'organisations réelles qui permettent d'illustrer l'expressivité du méta-modèle. Tout d'abord, nous présentons le cas Bolet, ensuite nous présentons le cas Seita, et nous terminons par le cas Touch.

2.4.1 – Le cas Bolet

Le cas Bolet est un cas d'étude de l'ouvrage universitaire « Sociologie des Organisation » [Bernoux 1985] [Mailliard, 2008]. Il décrit la tentative d'introduire davantage de rationalisation dans une entreprise jusqu'alors organisée selon un mode paternaliste. Le passage vers un mode de gestion « moderne » se traduit par la volonté d'introduire une nouvelle machine plus performante, quantitativement et qualitativement, afin de répondre aux exigences des clients et d'anticiper l'exploitation des possibilités de ce nouvel équipement par la concurrence. En terme de réorganisation du travail, les ouvriers de l'atelier de montage seraient amenés à perdre la maitrise de leurs compétences et surtout à se reconvertir vers un travail beaucoup plus standardisé.

L'entreprise Bolet illustre les difficultés du changement organisationnel toujours soumis à des stratégies de freinages. La SAO permet d'expliquer assez facilement ce type de phénomène. En général, le changement organisationnel induit une redistribution des zones d'incertitude. Les relations évoluent, certains acteurs seront alors amenés à perdre de leur pouvoir au profit de certains autres. Les acteurs capables de cette anticipation vont bien sûr se mobiliser afin de maintenir leur maîtrise sur les relations qu'ils contrôlaient jusqu'alors.

Plus concrètement, le cas Bolet peut être analysé par un modèle contenant quatre acteurs :

- *CA* : le Chef d'Atelier est un acteur historique, un ancien compagnon du fondateur de l'entreprise et un ancien ouvrier qui est écouté par le Père. Respecté par les ouvriers, il semble bien représenter les enjeux du personnel de production. Il est de plus en mesure de diriger effectivement les ateliers. Il n'y a donc pas lieu de le distinguer du personnel de production. Son autonomie réside dans la possibilité de déterminer l'investissement de l'atelier dans l'activité de production, et de plus ou moins bien appliquer les prescriptions du bureau d'étude.
- *Jean-BE* : Jean, le fils du patron, est le commercial de l'entreprise. Il est l'initiateur de la proposition d'introduire une nouvelle machine. Ses relations avec le bureau d'étude (BE) sont fonctionnelles. Elles consistent à adapter la production aux exigences qualitatives et quantitatives des clients. Les membres du bureau d'étude, tout comme Jean, ont fait des études supérieures. Ils sont fonctionnellement à des places différentes dans l'entreprise, mais sont solidaires par leurs intérêts commun à rationaliser les méthodes de production et leur même culture professionnelle. l'initiative de Jean relève plutôt des fonctions du BE et nous pouvons faire de Jean et du BE un seul et même acteur collectif. Jean détermine, à partir de ses relations avec les clients, les impératifs commerciaux et veut qu'ils soient pris en compte. Le BE établit la tâche de production prescrite, le type de cette production, son autonomie réside donc dans sa capacité à déterminer des normes et méthode de production.
- *André* : André est le fils aîné du patron et il est le responsable de la production. De par sa position institutionnelle, il est une sorte de « marginal sécant » situé dans les deux sous-systèmes qui s'affrontent (BE et CA). Il est à l'origine d'une réorganisation de l'entreprise et de la mise en place du BE. Fonctionnellement, c'est de lui que dépend la mise en place de la machine. Il est en position d'arbitre entre le BE et la Production. Il peut dans une certaine mesure contrôler les prescriptions proposées par le BE, c'est à dire les accepter ou non selon qu'elles sont plus ou moins contraignantes pour la production. De plus, il contrôle l'application de ces prescriptions par la production.
- *Père* : Le père Bolet préside l'entreprise dont il est le fondateur. Il maîtrise la décision

finale de d'introduire ou non la machine. Il ne semble pas avoir d'enjeux propres dans cet achat, si ce n'est le bon fonctionnement de l'entreprise qui passe par l'investissement et la « satisfaction » de chacun. D'autre part, on considère que ses intérêts sont essentiellement placés sur la satisfaction des autres.

Ces acteurs dépendent les uns des autres par l'intermédiaire des six relations suivantes :

- *La décision d'achat de la machine* : elle est contrôlée par le Père et tous en dépendent. Son échelle de valeur de l'état e st la suivante :

rejet = -10 ne prends pas position = 0 adoption = 10

- L'*application de la prescription* : elle est contrôlée par le CA et tous en dépendent. Son échelle de valeur de l'état e st la suivante :

ne l'applique pas du tout = -10 l'applique intégralement = 10

- L'*investissement dans la production* : elle est contrôlée par le CA et tous en dépendent sauf Jean-BE. Son échelle de valeur de l'état e st la suivante :

freinage = -10 zèle = 10

- Le *contrôle de l'application de la prescription* : elle est contrôlée par André et tous en dépendent sauf Jean-BE. Son échelle de valeur de l'état e st la suivante :

aucun contrôle = -10 contrôle tatillon = 10

- *La nature de la prescription du travail* : elle est contrôlée par Jean-BE et tous en dépendent. Son échelle de valeur de l'état e st la suivante :

peu contraignante = -10 rationalité technique rigoureuse = 10

- *Le contrôle de la nature de la prescription* : elle est contrôlée par André, et tous en dépendent sauf le CA. Son échelle de valeur de l'état e st la suivante :

aucun contrôle = -10 contrôle tatillon = 10

Une analyse du fonctionnement de l'entreprise [Maillard, 2008] conduit à définir les enjeux, et les fonctions d'effet comme indiqué dans les tableaux 2.5 et 2.6. Dans cette version du cas Bolet, chacun met 1 point de solidarité sur lui-même et 0 sur les autres.

	CA	Père	André	Jean-BE
decision-achat	4	**1**	1	4
application-prescription	**1**	1	1.5	2
investissement-dans-production	**2**	5	3	0
contrôle-application-prescription	2	1	**1.5**	0
nature-prescription	1	1	1.5	**2**
contrôle-nature-prescription	0	1	**1.5**	2

Tableau 2.5. *Les enjeux des acteurs sur les relations (les contours renforcés indiquent le contrôleur de la relation).*

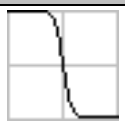
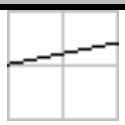
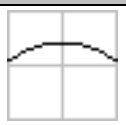
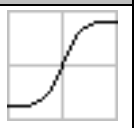
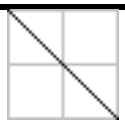
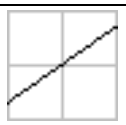
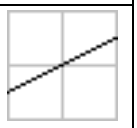
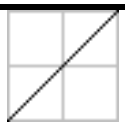
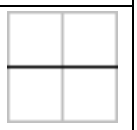
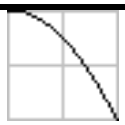
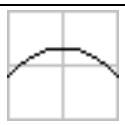
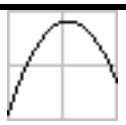
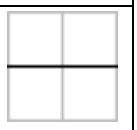
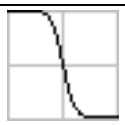
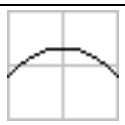
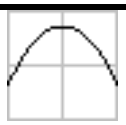
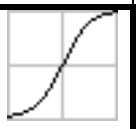
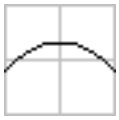
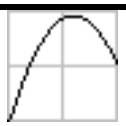
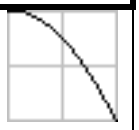
	CA	Père	André	Jean-BE
decision-achat				
application-prescription				
investissement-dans-production				
contrôle-application-prescription				
nature-prescription				
contrôle-nature-prescription				

Tableau 2.6. *Les fonctions d'effet des relations sur les acteurs (l'axe des x correspond à l'espace de comportement de l'état de la relation, l'axe des y à l'effet sur l'acteur).*

L'état d'une organisation est un vecteur de l'état de chaque relation, chaque état de l'organisation correspond à une configuration selon laquelle elle peut fonctionner. Puisque nous sommes intéressés à comprendre les comportements réels des acteurs de l'organisation, nous pouvons explorer de façon analytique l'ensemble de tous les états possible de l'organisation. Parmi eux, certains sont d'un grand intérêt parce qu'ils maximisent un critère donné, par exemple l'équilibre de Nash, les optima de Pareto, ou les extrema de la satisfaction d'un acteur ou de la somme de leurs satisfactions. L'environnement SocLab calcul ces états particuliers et donne des détails de chacun d'eux.

A titre d'exemple, le tableau 2.7 montre les états de l'organisation qui maximisent ou minimisent les satisfactions de chacun des acteurs (en ligne), et les satisfactions que cet état fournit à chacun (en colonne). Par exemple, le maximum de satisfaction que le chef d'atelier (CA) peut obtenir est de 100, et dans ce cas la satisfaction globale est de 87 tandis que la satisfaction du Père est de 42.

		CA	Père	André	Jean-BE	Globale
Max satisfaction	CA	**100**	42	-13,9	-41,1	87
	Père	-20,5	**65,8**	75,8	51,4	172,5
	André	17,3	63,5	**80,4**	14,9	176,1
	Jean-BE	-60	-49,5	-52,6	**81,1**	-81
	Globale	63,3	62,9	74,8	2,2	**203,2**

Min satisfaction	CA	**-100**	-49,5	-44,2	81,1	-112,6
	Père	60	**-57,5**	-73,6	-41,1	-112,2
	André	60	-57,5	**-73,6**	-41,1	-112,2
	Jean-BE	60	-57,5	-58,6	**-81,1**	-137,2
	Globale	-55,4	-55,1	-46,8	-32,1	**-189,4**

Tableau 2.7. *Les satisfactions des acteurs dans des états particuliers du cas Bolet qui maximisent ou minimisent les satisfactions des acteurs ou la satisfaction globale.*

2.4.2 – Le cas Seita

Le « monopole industriel » est une grande entreprise industrielle française qui détient, dans les années 1960, le monopole de l'état dans le secteur du tabac. Dans le phénomène bureaucratique, M. Crozier analyse de façon très approfondie à la fois la gestion des usines et le fonctionnement des ateliers de productions ([Crozier, 1963], pages 67-174, et 186-214]). Toutes les usines sont organisées selon le même schéma, et elles rencontrent les mêmes problèmes, et il en est de même au niveau des ateliers.

Chaque atelier réunit un chef d'atelier, des ouvriers de production et des ouvriers d'entretien. Le chef d'atelier supervise l'atelier : il s'occupe de la comptabilité de la production de l'atelier et de chaque ouvrier de production, de l'approvisionnement et de l'utilisation des matières premières. Il veille au bon fonctionnement de l'atelier et décide de redéploiement des ouvriers de production en cas de vacances de postes. Les ouvriers de production (entre 60 et 120 dans chaque atelier) sont surtout des femmes, peu qualifiés, et sous le commandement du chef d'atelier. Les ouvriers de maintenances sont des travailleurs bien qualifiés, ils dépendent d'un ingénieur technique qui ne fait pas partie de l'atelier, et chacun par eux est responsables de l'installation et la maintenance de trois machines. L'organisation de l'atelier et la répartition des emplois des ouvriers de production sont régis par des règles très claires et impersonnelles : tout le monde sait ce qu'il faut faire et comment, et rien n'est laissé à la discrétion des individus.

L'étude détaillée de M. Crozier a montré, dans les ateliers des 30 usines de la société, le même type de relations dysfonctionnelles entre les trois acteurs :

- Entre le chef d'atelier et les ouvriers de production : il y a peu de relations, et elles sont en général assez bonnes relations, même si (ou parce que) les ouvriers de production ne reconnaissent pas au chef son rôle de direction.
- Entre les ouvriers de production et ceux de maintenance : les relations sont conflictuelles, mais ne s'expriment pas franchement : les ouvriers de production accusent ceux de maintenance de ne pas faire le nécessaire pour réparer les machines rapidement, mais ils sont plutôt soumis. Les ouvriers de maintenance considère ceux de production comme leurs subordonnés, et se permettent d'interférer dans leur travail ; ils disent que les ouvriers de production sont négligents et ne travaillent pas assez.
- Entre le chef d'atelier et les ouvriers de maintenance : les relations sont clairement hostiles ; les ouvriers de maintenance critiquent les compétences du chef d'atelier d'une façon très agressive et ils lui dénient toute importance. Les chefs d'ateliers sont plus réservés dans leurs critiques des ouvriers de maintenance, tout en dénonçant qu'ils abusent de leur pouvoir sur l'entretien des machines.

En résumé, le cas Seita est un modèle contenant trois acteurs – le chef d'atelier « CA », le groupe des ouvriers de production « ProdE », et le groupe des ouvriers de maintenance « MainE » – et quatre relations – CA contrôle l'application des règles formelles « règles », ProdE contrôle son investissement dans les activités productives « production », et MainE contrôle d'une part la

maintenance des machines « maintenance » et d'autre part son agressivité envers CA « pressions ». L'interprétation de l'espace des comportements des relations est donnée dans le tableau 2.8.

Relations	Valeurs	Le comportement du contrôleur de la relation
Règles	Négatives	CA interprète les règles formelles en fonction de la situation concrète au mieux du fonctionnement de l'atelier
	Positives	CA applique les règles aveuglement
Production	Négatives	ProdE manquent d'intérêt pour leur travail
	Positives	ProdE s'investissent dans leur travail, et font leur mieux
Maintenance	Négatives	MainE s'occupent des machines à leur convenance, de façon imprévisible
	Positives	MainE règlent et réparent les machines de façon rapide et efficace
Pressions	Négatives	MainE expriment une forte agressivité envers le CA
	Positives	MainE coopèrent avec le CA

Tableau 2.8. *Les interprétations des états des relations - les comportements du contrôleur de la relation.*

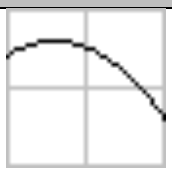
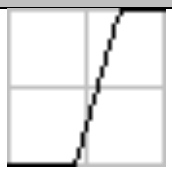
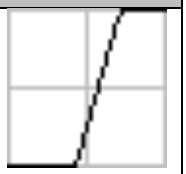
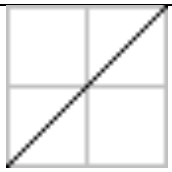
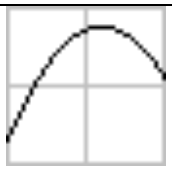
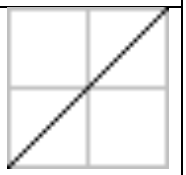
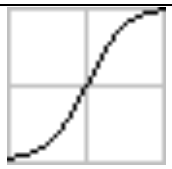
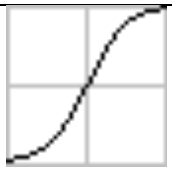
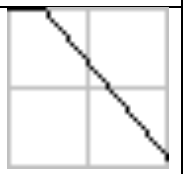
Les enjeux des acteurs sur les relations et les solidarités de chaque acteur sur les autres peuvent être évalués comme indiqués dans le tableau 2.9.

Enjeux	CA	ProdE	MainE
Règles	**3**	3	4
Production	3	**2**	1
Maintenance	3	5	**4**
Pressions	1	0	**1**

Solidarités	CA	ProdE	MainE
CA	0,9	0	-0,1
ProdE	0	1	0
MainE	-0,15	0	0,85

Tableau 2.9. *Les enjeux des acteurs sur les relations dont ils dépendent (les contours renforcés indiquent le contrôleur de chaque relation), et les solidarités de chaque acteur (en ligne) sur les autres (en colonne).*

Les fonctions d'effet des relations sur les acteurs sont présentées dans le tableau 2.10, ainsi que les contraintes entre quelques relations dans le tableau 2.11.

	CA	ProdE	MainE
Application des Règles			
Investissement dans la Production			
Qualité de la Maintenance			

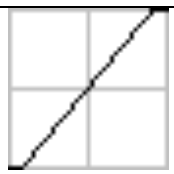
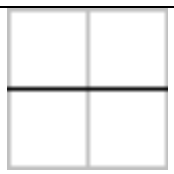
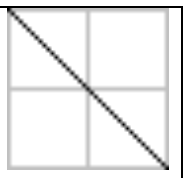
Pressions sur le Chef d'Atelier			

Tableau 2.10. *Les fonctions d'effet des relations sur les acteurs (l'axe des x correspond à l'espace de comportement de la relation, l'axe des y à l'effet sur l'acteur).*

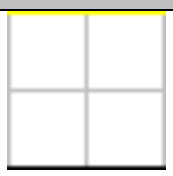
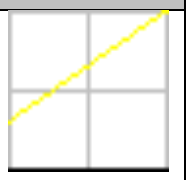
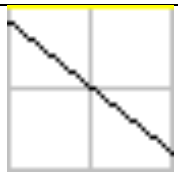
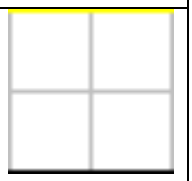
	Règles	Production
Maintenance		
Pressions		

Tableau 2.11. *Les contraintes de l'état d'une relation (en ligne) sur les bornes inférieures et supérieures de l'état d'une autre relation (en colonne). L'axe des x correspond à l'état de la relation de contraignante, l'axe des y aux bornes de la relation contrainte : l'état de la maintenance contraint la valeur maximale de l'état de la production, et l'état de la pression contraint la valeur minimale de l'état des règles.*

Le tableau 2.12 montre les états de l'organisation qui maximisent ou minimisent la satisfaction de chaque acteur (en ligne), et les satisfactions que cet état fournit à tous les acteurs (en colonne). Par exemple, le maximum de satisfaction que le chef d'atelier (CA) peut obtenir est de 85,2, et dans ce cas la satisfaction globale est de 28,8 tandis que la satisfaction de ProdE est de 20,2.

		CA	ProdE	MainE	Globale
Max Satisfaction	CA	**85,2**	20,2	-76,7	28,8
	ProdE	25,6	**93,0**	9,1	127,5
	MainE	-28,8	-16,2	**88,1**	43,2
	Globale	42,9	79,3	23,8	**146,1**
Min Satisfaction	CA	**-80,8**	-32,2	80,1	-33,0
	ProdE	-51,2	**-92,2**	8,5	-135
	MainE	33,2	4,2	**-84,7**	-47,4
	Globale	-51,2	-92,2	8,5	**-135**

Tableau 2.12. *Les satisfactions des acteurs dans les configurations du cas Bolet qui maximisent ou minimisent les satisfactions des acteurs ou la satisfaction globale.*

2.4.3 – Le cas Touch

Dans cette section, nous présentons le système d'action organisée en charge de la gestion d'une rivière, le « Touch », qui est précisément documenté dans [Baldet, 2012]. Le Touch est une rivière du sud-Ouest de la France qui coule dans le département de la Haute-Garonne en Midi-pyrénées. C'est un affluent de la rive gauche de la Garonne dans laquelle elle se jette au nord de Toulouse, ville d'un million d'habitants. Son parcours traverse 29 communes et Son bassin versant

englobe le territoire de 60 communes. Trois quarts de ces municipalités sont situées en amont et sont essentiellement des villages d'agricoles ou des petites villes. Le quart aval de ce cours d'eau pénètre dans un paysage urbain à proximité de l'agglomération toulousaine. Les municipalités en aval ont été atteintes par plusieurs épisodes d'inondations au cours des dernières décennies. Elles considèrent que les municipalités en amont ne coopèrent pas assez, et elles ont essayé de se protéger en construisant des digues qui, même si elles sont coûteuses, ne sont pas suffisantes pour éliminer le risque d'inondation. Au contraire, les municipalités en amont, fortement sous l'influence des agriculteurs, considèrent qu'elles ont pris leurs responsabilités dans la prévention des inondations en laissant certaines terres agricoles en friche en vue d'absorber l'excès d'eau en cas d'inondation. Le modèle de ce système d'acteur a été conçu à l'occasion du renouvellement du plan de prévention des risques d'inondation du Touch (PPRI), une obligation française depuis 1987 qui a été renforcée par la directive cadre Européenne sur l'eau (European Water Framework Directive, WFD 2000/60/EC).

Plus concrètement, le système d'action comprend dix acteurs qui sont impliqués dans la gestion de la rivière et ont un intérêt dans l'élaboration et dans les résultats :

- DDT : la direction départementale du territoire agit en tant que représentant de l'état et doit promulguer la nouvelle.
- ONEMA : l'office national de l'eau et des milieux aquatiques est l'organisme de référence pour les études et le suivi des questions regardant l'eau et les milieux aquatique.
- AEAG : l'agence de l'eau Adour-Garonne est l'autorité opérationnelle chargée des plans stratégiques. En tenant compte des exigences des différents usages de l'eau et de la protection des milieux aquatiques, il définit, supervise et finance la politique de l'eau au niveau du bassin.
- ARTESA : une association citoyenne d'agriculteurs riverains dans la zone amont. Ils sont propriétaires de terres inondables et, comme les riverains du Touch, ont le droit d'utiliser le fleuve et doivent maintenir ses rives.
- Municipalités_Amont : le groupe des 25 municipalités en amont dont la population est de 21 000 habitants.
- Municipalités_Aval : le groupe des 4 municipalités aval (75 000 habitants) qui sont concernés par chaque occurrence d'une catastrophe naturelle. En raison de des menaces d'inondation, ces municipalités doivent interdire toute construction sur une partie de leur territoire.
- SIAH : l'association intercommunale d'aménagement hydrolique est en charge de la gestion du Touch. Elle doit, avant tout, entretenir le lit et les rives de la rivière. Elle comprend des représentants des 29 municipalités riveraines et son responsable favorise la coopération entre les municipalités tout en se souciant du bon état écologique de la rivière.
- CR : le conseil régional peut apporter un soutien financier supplémentaire aux mesures de génie civil.
- CG : le conseil départemental peut aussi apporter un soutien financier supplémentaire.
- SOGREAH : un bureau d'étude spécialisé dans l'eau, l'énergie, et l'environnement, chargé des études techniques.

Les acteurs qui sont les plus engagés dans la négociation sont ARTESA, Municipalités_Amont, et Municipalités_Aval du point de vue de la population, et AEAG, SIAH, et CG du point de vue institutionnel. Tous ces acteurs ont des intérêts importants dans les résultats des discussions. Les autres acteurs sont plutôt périphériques.

Chacun des acteurs contrôle une relation qui résume ses moyens d'action dans la négociation :

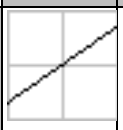
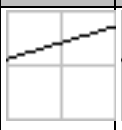
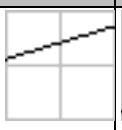
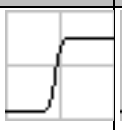
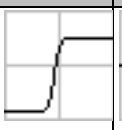
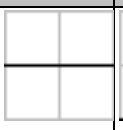
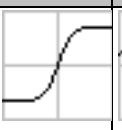
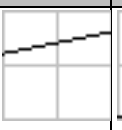
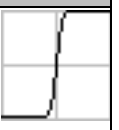
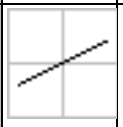
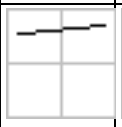
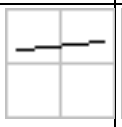
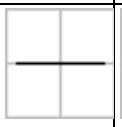
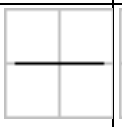
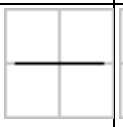
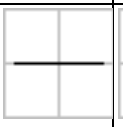
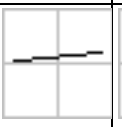
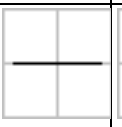
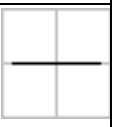
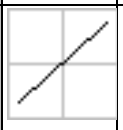
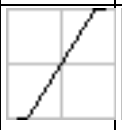
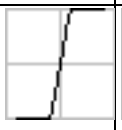
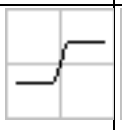
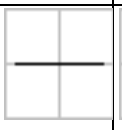
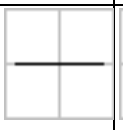
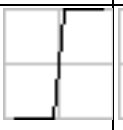
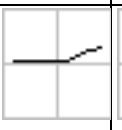
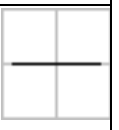
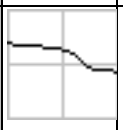
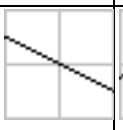
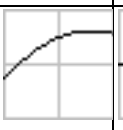
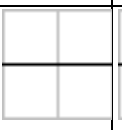
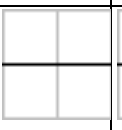
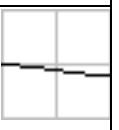
- *Validation* (entre -10 et 10) est la plus ou moins grande sévérité de la DDT pour l'approbation du plan de prévention proposé par SIAH. Cette validation se fait sur la base de critères techniques et écologiques.
- *Expertise* (entre -8 et 8) est le résultat d'une étude dont ONEMA est en charge. Il peut donner un effet positif ou négatif sur l'évaluation des travaux d'aménagement, basé principalement sur des critères écologiques.
- *Financement* (entre -8 et 8) est le financement provenant de l'AEAG qui peut payer jusqu'à 75% du coût total des travaux de si le projet en considéré comme écologique.
- *P*ression (entre -10 et 10) est l'action menée par ARTESA qui représente les propriétaires des terres inondables. Bien que ARTESA ne soit pas très concernée par les questions écologiques, elle a une excellente connaissance du terrain et se trouve souvent à argumenter contre ONEMA et AEAG.
- *Le contrôle de flux* (entre -8 et 8) est la capacité des municipalités en amont (Municipalités_Amont) à garder sur leur territoire une partie de l'eau qui inonde les communes en aval.
- *L'auto-financement* (entre -8 et 8) est la capacité des municipalités en aval (Municipalités_Aval) de financer les travaux de génie civil.
- *La gestion de la rivière* (entre -8 et 8) est l'activité du SIAH sur la gestion de la rivière : niveau bas signifie que l'association minimise son implication dans la maintenance de la rivière et un niveau élevé signifie que l'association est impliqué en essayant de prévenir les menaces provenant de la rivière.
- *Financement complémentaire du CR* (entre -8 et 8) est la participation financière du CR dans les projets d'aménagement.
- *Financement complémentaire du CG* (entre -8 et 8) est la participation financière du CG dans les projets : CG possède ses propres règles pour le soutien financier à un projet. Un niveau élevé de l'état de cette relation signifie des contraintes plus strictes (principalement écologiques) pour accorder le financement à un projet.
- *Études* (entre -8 et 8) est une étude réalisée par SOGREAH : une valeur positive signifie un résultat hydromorphologique de cette étude (approche écologique qui utilise la forme de la rivière pour essayer de prévenir les inondations) et une valeur négative signifie un résultat hydraulique de cette étude (construction des digues, ce qui est moins écologique).

Le modèle de la gestion du Touch analysé dans [Sibertin-Blanc *et al*, 2013] définit les enjeux des acteurs sur les relations (tableau 2.13), les fonctions d'effets des relations sur les acteurs (tableau 2.14), et les solidarités entre les acteurs (tableau 2.15).

	DDT	ONEMA	AEAG	ARTESA	Municipalités Amont	Municipalités Aval	SIAH	CR	CG	SOGREAH
Validation	**4**	2	1	0,5	0,5	0	0,5	1	2	1,5
Expertise	1	**3**	0,5	0	0	0	0	0,5	0	0
Financement	1	1,5	**4**	0,5	0	0	2	2	1,5	0
Pression	0,5	0,5	1	**4**	1	1	0,5	0	0	1
Le contrôle de flux	0	1,5	1	2	**4**	2	2	0	0	1,5

Auto-financement	0,5	0	0	1	1,5	**4**	2	0,5	0,5	0
La gestion de la rivière	2	1,5	1,5	1,5	2,5	2,5	**3**	2,5	2,5	2
Finan. compl. du CR	0,5	0	0,5	0	0	0	0	**3**	0,5	0
Finan. compl. du CG	0,5	0	0,5	0	0	0	0	0,5	**3**	0
Études	0	0	0	0,5	0,5	0,5	0	0	0	**4**

Tableau 2.13. *Les enjeux des acteurs sur les relations dont ils dépendent (les contours renforcés indiquent le contrôleur de chaque relation).*

	DDT	ONEMA	AEAG	ARTESA	Municipalités Amont	Municipalités Aval	SIAH	CR	CG	SOGREAH
Validation										
Expertise										
Financement										
Pression										
Le contrôle de flux										
Auto-financement										
La gestion de la rivière										
Finan. Com. du CR										
Finan. Com. du CG										

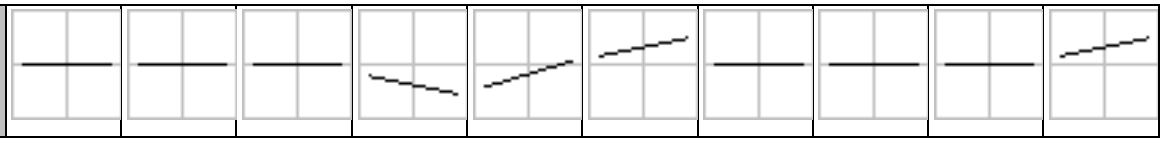
Études	

Tableau 2.14. *Les fonctions d'effet des relations sur les acteurs (l'axe des x correspond à l'espace de comportement de la relation, l'axe des y à l'effet sur l'acteur).*

	DDT	ONEMA	AEAG	ARTESA	Municipalités Amont	Municipalités Aval	SIAH	CR	CG	SOGREAH
DDT	0,5	0,05	0,05	0,1	0,1	0	0,1	0	0	0
ONEMA	0,05	0,6	0,15	-0,05	0,05	0	0,2	0	0	0
AEAG	0,05	0,1	0,7	-0,05	0	0	0,2	0	0	0
ARTESA	0,05	0	0	0,8	0,2	-0,1	0,05	0	0	0
Municipalités Amont	0,2	0	0	0,3	0,7	-0,1	0	0	0	-0,1
Municipalités Aval	0,1	0,15	0,3	-0,2	-0,1	0,7	0	0	0	0,05
SIAH	0,1	0	0,05	0	0,1	0,1	0,65	0	0	0
CR	0	0,05	0,1	0	0	0,05	0,1	0,7	0	0
CG	0	0,1	0,2	0	0	0	0	0,1	0,6	0
SOGREAH	0	0,1	0,15	-0,1	-0,05	0	0,2	0	0	0,7

Tableau 2.15. *Les solidarités de chaque acteur (en ligne) sur les autres (en colonne).*

Les tableaux 2.16 et 2.17 montre les états de l'organisation qui maximisent ou minimisent les satisfactions de chaque acteur (en ligne), et les satisfactions que cet état fournit à tous les acteurs (en colonne). Par exemple, le maximum de satisfaction que DDT peut obtenir est de 63,2, et dans ce cas la satisfaction globale est de 554 tandis que la satisfaction de ARTESA est de 77,4.

	DDT	ONEMA	AEAG	ARTESA	Municipalités Amont	Municipalités Aval	SIAH	CR	CG	SOGREAH	Globale
DDT	**63,2**	59,1	68,3	77,4	72,2	-3,7	58,9	52,3	55,5	50,6	554
ONEMA	47,8	**68,1**	79,5	-24,3	-24,9	70,8	61,6	58,4	57,6	73,3	468
AEAG	48,4	67,3	**82,6**	-5,3	-31,7	77	55,5	60,3	60,7	74,7	489
ARTESA	61,9	53,3	62,4	**84,4**	75,9	-26,9	46,5	49,3	55	38,3	500
Municipalités Amont	59	48,2	56,1	80,7	**80**	-15,6	34,3	45,2	52,5	25,9	466
Municipalités Aval	39,8	64,3	74,2	-52,3	-49,3	**109,9**	36,6	56,7	57,3	67,5	405

SIAH	57,3	65,7	77,4	24,3	26,7	59,9	**70,8**	58,6	56,4	62,3	560
CR	49,6	66,7	81,7	-1,3	-26	73,4	59,4	**61**	60,7	73,9	499
CG	47,9	61,3	76,4	26,4	-5,3	29	26,3	52,3	**63,8**	55,8	434
SOGREAH	40,7	67,5	78,1	-44,9	-53,4	87,4	54,5	55,6	55,1	**78,9**	420
Globale	60,4	63,2	74	51	43,4	44,3	70	57,8	56,9	57,6	**579**

Tableau 2.16. *Les satisfactions des acteurs dans des configurations du cas Touch qui maximisent les satisfactions des acteurs ou la satisfaction globale.*

	DDT	ONEMA	AEAG	ARTESA	Municipalités Amont	Municipalités Aval	SIAH	CR	CG	SOGREAH	Globale
DDT	**-40,9**	-9,5	-39,3	-80	-77,8	55,7	-49,1	-36,9	-18,8	-25	-320
ONEMA	-20	**-23,2**	-54,4	53,5	50,3	-67,5	-45,2	-46,3	-22,1	-52	-227
AEAG	-28,6	-20,7	**-56,8**	-14,3	16,2	-45	-48,5	-47	-23,6	-43,7	-312
ARTESA	-40,8	-9,3	-39,2	**-80,9**	-78	57,5	-48,7	-36,9	-18,8	-16	-311
Municipalités Amont	-29,2	6,5	-18,1	-68,4	**-83,1**	50,6	-11,9	-22	-7,4	25,6	-157
Municipalités Aval	-8	-7,4	-33,4	66,4	48,7	**-75**	-8,2	-32	-11,2	-11,9	-72
SIAH	-38,5	-12,1	-42,9	-54,1	-50,9	4,9	**-61,7**	-43,6	-17,7	-30,6	-347
CR	-28,6	-20,7	-56,8	-14,3	16,2	-45	-48,5	**-47**	-23,6	-43,7	-31,2
CG	-31	-18,1	-53,2	-40,2	-10,7	5,8	-35,8	-40,3	**-24,7**	-38,1	-286
SOGREAH	-19,5	-23,2	-54,4	53,6	50,5	-67,5	-45,1	-45,2	-22	**-52**	-225
Globale	-38,5	-12,1	-42,9	-54,1	-50,9	4,9	-61,7	-43,6	-17,7	-30,6	**-347**

Tableau 2.17. *Les satisfactions des acteurs dans des configurations du cas Touch qui minimisent les satisfactions des acteurs ou la satisfaction globale.*

2.5 – Le jeu Social

Un contexte d'interaction défini en termes d'acteurs et de relations institue un « jeu social » dans lequel chaque acteur cherche à manipuler les relations qu'il contrôle afin d'obtenir un niveau de satisfaction acceptable.

Pour obtenir un niveau de satisfaction satisfaisant, chaque acteur ajuste son comportement, c'est-à-dire modifie l'état des relations qu'il contrôle à chaque pas du jeu ; ce faisant, il modifie la satisfaction des acteurs qui en dépendent, y compris sa propre satisfaction. Plus précisément, à chaque étape du jeu, chaque acteur agit sur les relations qu'il contrôle en modifiant leurs états : soient $(e_{r1}, \ldots, e_{rm})$ l'état de l'organisation et $(c_r)_{contrôle(r) = a}$ les modifications choisies par un acteur a, avec $(c_r + e_r) \in [-10 ; 10]$. Une fois que chaque acteur a effectué ses modifications, le jeu passe dans un nouvel état défini par :

$$Transition : [-10;10]^m \times [-10;10]^m \rightarrow [-10;10]^m$$

$$((e_{r1}, \ldots, e_{rm}), (c_{r1}, \ldots, c_{rm})) \mapsto (c_{r1} + e_{r1}, \ldots, c_{rm} + e_{rm})$$

Chaque acteur sélectionne son action en fonction de sa satisfaction courante qui correspond à sa fonction d'utilité. Le jeu se termine quand un état stationnaire est atteint : les acteurs ne cherchent plus à changer l'état des relations qu'ils contrôlent, parce qu'ils sont satisfaits du niveau de satisfaction que l'état courant de l'organisation leur procure. Le SAC est alors régulé et peut fonctionner ainsi, chacun des acteurs ayant trouvé un comportement qui est satisfaisante pour lui-même et pour les autres.

Il s'agit bien d'un jeu au sens de [Morgenstern *et al.*, 1953] puisque chaque acteur contribue à déterminer l'état du SAC et en retire une certaine « utilité », sa satisfaction (le jeu peut aussi se définir en considérant l'influence, de chaque acteur, ou sa capacité d'action ou son pouvoir). Le jeu social se distingue cependant des jeux tels qu'ils sont considérés en économie, puisqu'il ne s'intéresse pas aux gains, éventuellement cumulés, que chaque acteur peut obtenir à chaque étape ; il s'intéresse aux conditions même d'existence du jeu, c'est-à-dire à la possibilité que les acteurs trouvent une façon d'adapter leurs comportements les uns aux autres qui permette à l'organisation de perdurer en satisfaisant raisonnablement ce qui fait sa raison d'être. Si le jeu parvient à un état stationnaire, c'est que chaque acteur a trouvé un comportement qui lui convient et le SAC peut perdurer.

2.6 – La rationalité limitée des acteurs sociaux

Selon la SAO, le comportement des acteurs sociaux est stratégique, c'est-à-dire finalisé : chacun utilise les relations qu'il contrôle comme leviers d'actions afin d'influencer le comportement des autres pour en obtenir une satisfaction qui lui convienne. C'est cette dimension stratégique des comportements des acteurs qui leur assure une relative stabilité, et donne lieu au phénomène de régulation : les acteurs se comportent comme s'ils obéissaient à des règles qu'en fait ils instaurent eux-mêmes.

Pour autant, ce comportement n'est pas nécessairement délibéré et il s'exerce dans le cadre d'une rationalité limitée [Simon, 1982] qui, dans un contexte donné, recherche une solution satisfaisante tout en prenant en compte le coût cognitif de cette recherche et l'imperfection des informations dont l'acteur dispose, tant sur l'état que sur la structure du contexte auquel il est confronté :

- Il ne s'agit pas d'une rationalité substantive qui évaluerait selon un critère bien défini l'ensemble préalablement connu des alternatives possibles, mais d'une rationalité procédurale par laquelle, en fonction du contexte dans lequel il se trouve, l'acteur cherche dynamiquement des alternatives possibles, les évalue et s'arrête dès qu'une solution satisfaisante est trouvée.
- Si les objectifs d'un acteur social sont potentiellement flous et contradictoires, ils n'en déterminent pas moins certaines aspirations que l'acteur cherche à réaliser ; dans le domaine du social, « chacun, dans sa sphère, se rend vaguement compte du point extrême jusqu'où peuvent aller ses ambitions et n'aspire à rien au-delà » ([Durkheim, 1897], p. 23) ; le niveau des aspirations d'un acteur, qui détermine les situations dans lesquelles il s'estimera satisfait, est largement dépendant du contexte et n'est donc pas prédéfini.
- Les décisions sont prises sur la base d'informations incomplètes, voire erronées, tant sur l'état que sur la structure du contexte d'interaction, ce qui entraîne une incertitude sur l'évaluation *a priori* et sur l'effet *a posteriori* des alternatives ; cet aspect renvoie à une dimension essentielle des processus sociaux, à savoir leur l'opacité [Roggero *et al.*, 2003].
- Les humains ont des capacités cognitives limitées qui les empêchent d'évaluer parfaitement toutes les alternatives et ils tiennent compte du coût de la recherche d'une nouvelle solution (les bébés de six mois [Coisne, 2005] et les personnes déficientes [Sacks,

1988] ont un comportement social).

- Enfin, les jeux sociaux sont des jeux à somme non nulle : la satisfaction qu'un acteur octroie à un autre ne l'est pas nécessairement à son détriment, notamment si l'on prend en compte la réciprocité des échanges de comportement entre les acteurs [Mailliard et al., 2010]. Les comportements sociaux sont donc globalement coopératifs, cette coopération étant nécessaire au bon fonctionnement et à la perpétuation de l'organisation, ce à quoi chaque acteur est attaché, sous l'hypothèse qu'il a de bonnes raisons de faire partie de cette organisation.

Chapitre 3 – État de l'art sur l'apprentissage

D'après [De Ketele *et al.*, 1988], l'apprentissage est un processus systématiquement et intentionnellement orienté vers l'acquisition de certains savoir, savoir-faire, savoir-être, et savoir-devenir par l'observation, l'imitation, l'essai, et la répétition. Dans notre contexte, il s'agit de permettre à un acteur artificiel, nommé aussi apprenant ou agent, d'apprendre comment se comporter, c'est-à-dire d'associer à des situations rencontrées des actions à réaliser. Ce processus cognitif est essentiel chez les êtres humains sous la forme de l'apprentissage naturel dès la naissance. Il permet aussi à un animal d'utiliser son expérience passée pour assimiler l'organisation de son environnement et les conséquences de ses propres actions, et pour s'y adapter. L'apprentissage automatique permet à une machine, que ce soit un programme ou un robot, de découvrir un environnement, d'en construire un modèle interne, de prévoir le comportement et d'analyser l'environnement grâce à ce modèle, et d'effectuer des tâches complexes qui requièrent de l'intelligence, comme l'aide à la décision, la conduite de robots, l'exploration de grandes bases de données, etc.

Plusieurs approches performantes et puissantes ont été développées pour résoudre le problème de l'apprentissage par des machines. On peut distinguer trois formes d'apprentissage :

- *L'apprentissage supervisé* : c'est une technique visant à faire acquérir automatiquement des règles à partir d'un ensemble prédéfini d'exemples. Par exemple, un acteur qui décide de manger des champignons en évitant de mourir, doit être capable de discerner les champignons comestibles et les champignons empoisonnés. Pour cela, il doit tout d'abord collecter les données : cueillette de champignons et étiquetage des champignons cueillis par un pharmacien. Ensuite il doit analyser ces données pour les identifier : mesurer une série d'indicateurs (hauteur/largeur, couleur du pied/chapeau, odeur, poids, …). Après l'analyse des indicateurs, il doit créer des catégories et des règles : par exemple, si un champignon est rouge ou jaune alors il est empoisonné. Ensuite, il doit évaluer les résultats : tester la règle apprise sur une nouvelle cueillette. Et finalement il doit déployer la solution.

  Les algorithmes d'apprentissage supervisé permettent d'apprendre à partir d'exemples. L'objet de l'apprentissage est de copier un comportement déjà validé. Ce genre d'apprentissage n'est valable que si l'entrée (les exemples) et la sortie (le comportement désiré) sont connues à l'avance.

  L'apprentissage supervisé est souvent utilisé pour faire des prévisions : un acteur ayant un nombre déterminé d'observations, devra être capable de classer une nouvelle observation parmi celles dont il dispose. Ce type d'apprentissage est par exemple utilisé dans les domaines du diagnostic médical (malade/sain) ou de la reconnaissance de caractères.

- *L'apprentissage non-supervisé* : avec l'apprentissage supervisé, on a supposé que l'ensemble des classes et solutions était défini ; ce n'est pas toujours le cas. Un enfant apprend à « catégoriser », c'est-à-dire à associer un objet à une classe alors même que la classe en question n'a pas de définition bien précise : par exemple, comment définir ce qui relève de la catégorie « chaise » ? Même lorsque l'espace des classes est bien défini, il est parfois nécessaire que l'acteur ait à les redécouvrir au cours du processus d'apprentissage. On peut par exemple espérer qu'un enfant mis en présence de mammifères (à pattes), de reptiles, d'oiseaux et de poissons saura faire les regroupements adéquats et apprendra du coup les classes et le moyen de les différencier. On parle dans ce cas d'*apprentissage non supervisé*, autrement dit, sans professeur.

  Cette technique diffère donc de l'apprentissage supervisé du fait que l'acteur dispose initialement d'une base de données d'apprentissage vide qu'il va faire évoluer au gré de ses

expérimentations. Le but de l'apprentissage non-supervisé est de trouver un lien entre les entrées.

Ce type d'apprentissage est souvent utilisé pour faire de la classification (regroupement des exemples similaires en agrégats qui caractérisent les différentes classes), pour détecter des règles d'association (analyser les relations entre les variables), etc.

- *L'apprentissage par renforcement* : c'est une technique permettant à un acteur d'apprendre à partir du succès et des échecs de ses actions. Un paradigme classique pour présenter les problèmes d'apprentissage par renforcement consiste à considérer un acteur autonome, plongé au sein d'un environnement, et qui doit prendre des décisions en fonction de son état courant et d'un objectif à atteindre. En retour, l'environnement lui procure une évaluation (appelée renforcement) scalaire de ce qu'il a fait, qui peut être positive ou négative. L'acteur cherche, au travers d'expériences itérées, un comportement, ou processus de décision des actions à entreprendre dans chaque cas (appelé stratégie ou politique, qui est une fonction associant à l'état courant l'action à exécuter) optimal, au sens qu'il maximise la somme des renforcements au cours du temps.

  Dans la plupart des cas, le comportement désiré est inconnu bien que le but de l'apprentissage soit bien défini. Par exemple, dans un jeu d'échecs, le comportement désiré est inconnu car on ne sait pas comment l'adversaire va réagir, bien que le but général soit clair : battre l'adversaire.

  L'apprentissage par renforcement fait usage du renforcement différé ; cela signifie que non seulement les renforcements immédiats, mais aussi les renforcements futurs (après un certain nombre d'étapes) sont importants pour évaluer le comportement et évoluer : par exemple, au jeu d'échecs, la sacrifice d'une pièce est parfois nécessaire pour obtenir une meilleure position.

3.1 – Les approches à exclure

Il ne s'agit pas de faire un état de l'art sur toutes les approches de l'apprentissage, mais plutôt d'en décrire quelques unes qui nous semblent importantes à mentionner. Tout d'abord, chaque approche sera présentée brièvement. Ensuite quelques articles pionniers ou récents dans le domaine seront cités. Et finalement, les raisons pour lesquelles cette approche n'est pas applicable dans notre cas seront expliquées.

3.1.1 – L'apprentissage par induction

L'apprentissage par induction est une forme d'apprentissage non-supervisée qui consiste à créer des lois générales à partir de l'observation de faits particuliers sur une base probabiliste. L'idée de départ de l'induction est que la répétition d'un phénomène augmente la probabilité de le voir se reproduire : par exemple, si une pomme qui se détache de son arbre tombe sur le sol, on peut établir une loi générale à propos de la gravitation universelle mais avec une probabilité ou une certitude très faible ; si ensuite, on observe que toutes les pommes et tous les corps que l'on voit tomber tombent de la même façon, alors la probabilité de la loi augmentera jusqu'à devenir une quasi certitude. En d'autres termes, cette forme d'apprentissage peut être formalisée par un jeu entre un environnement produisant différents exemples possibles, et un acteur qui utilise un principe inductif lui indiquant quelle hypothèse causale il doit choisir étant donnés les exemples d'apprentissage. Par exemple, tout le monde sait distinguer un corbeau d'un canard : leur différences de couleur, de cri, de vitesse d'envol, de silhouette … nombre d'attributs les différencient ; toute personne qui observe l'un de ces volatiles peut lui donner son nom pratiquement sans erreur. Pourtant, personne n'a certainement jamais vu tous les corbeaux ni tous les canards existants. Mais à partir d'observations en nombre limité, chacun a appris à les distinguer.

[Robinet *et al.*, 2008] ont introduit un modèle cognitif de l'apprentissage inductif de concepts de façon non-supervisée basé uniquement sur le « principe de simplicité », principe cognitif fondamental permettant de rendre compte de la façon dont l'être humain est capable d'apprendre des concepts à partir d'exemples. Ce principe énonce que, parmi toutes les explications cohérentes relatives à un jeu de données, chaque individu privilégie la plus simple.

L'apprentissage par induction est une forme d'apprentissage qui fonctionne très bien lorsqu'elle est bien encadrée par des exemples de bonne qualité. Il faut un espace d'hypothèses suffisamment riche pour pouvoir approcher la fonction cible d'assez près, mais il ne faut pas qu'il le soit trop, sous peine de conduire à des hypothèses bonnes sur les données d'apprentissage, mais mauvaises en généralité. Par exemple, si j'observe une seringue remplie d'air que je peux compresser et étirer, je peux en induire que l'air et les gaz en général sont compressibles. Par contre, si un enfant observe une plume et une roche qui ne tombent pas à la même vitesse dans l'air, il pourra induire que les objets lourds tombent plus vite ; ce qui est faux.

Dans notre contexte de rationalité limitée des acteurs sociaux, où les décisions sont prises sur la base d'informations incomplètes, voire erronées, tant sur l'état que sur la structure du jeu, l'application de l'induction n'est pas suffisamment précise. Par exemple, si un acteur perçoit une augmentation plus importante de sa satisfaction provenant d'une relation plutôt que d'une autre, il pourra induire que la première est plus profitable que la deuxième. Cette hypothèse n'est pas toujours vraie ; la variation de l'impact d'une relation sur un acteur dépend de l'enjeu que cet acteur a mis sur cette relation, mais aussi de l'action effectuée par l'acteur qui contrôle la relation : si le contrôleur est en mode d'exploration, il va choisir des actions plus énergiques, ce qui implique une variation importante. Par contre, si le contrôleur exploite, il va choisir des actions plus fines, ce qui impliquera une variation faible.

3.1.2 – Les approches logiques inductives

Les approches logiques inductives sont des approches d'apprentissage par induction qui utilisent les techniques de la programmation logique. Elles cherchent à réaliser l'apprentissage de formules de la logique des prédicats à partir d'exemples et de contre-exemples. En d'autres termes, un programme de PLI (Programmation Logique Inductive) part de situations particulières et mène à des inférences, c'est-à-dire à des concepts à partir d'observations et de propositions. Par exemple, à partir de la description des liens de parenté dans quelques familles (Jean est le père de Pierre, Paul est le père de Jean, Paul est le grand-père de Pierre, etc), un programme de PLI doit être capable de trouver une formule du type « Pour tous les x et y tels que x est le grand-père de y, il existe un z tel que x est le père de z et z est le père de y ». La PLI a deux caractéristiques fortes : d'abord, le langage de représentation des hypothèses est très bien défini mathématiquement et algorithmiquement. La notion de généralisation peut alors être utilisée en cohérence avec les outils de la logique, comme la démonstration automatique. Ensuite, du fait de la richesse de ce langage, la combinatoire de l'apprentissage est très grande : il s'agit d'explorer un espace immense en faisant constamment des choix qu'il sera difficile de remettre en question. C'est pourquoi il est important en PLI de bien décrire les biais d'apprentissage qui limitent cette exploration.

Ce type d'apprentissage est plutôt appliqué dans les domaines de classifications des données et l'apprentissage des concepts. Par exemple, [Chelghoum *et al.*, 2006] propose une approche basée sur la PLI qui utilise deux notions : la première consiste à matérialiser les interactions spatiales dans des tables de distances, ramenant ainsi la fouilles de données spatiales à la fouille de données multi-tables. La seconde transforme les données en logique du premier ordre et applique ensuite la PLI. [Fromont *et al.*, 2006] propose un algorithme d'apprentissage de règles provenant de données multi-sources par PLI qui s'appuie sur des apprentissages mono-sources.

Comme le langage de description des concepts est riche, la PLI peut être appliquée à un grand nombre de situations. En pratique, les algorithmes permettent d'apprendre des concepts opératoires

dans des domaines aussi variés que le traitement de la langue naturelle, la chimie, le dessin industriel, la fouille de données, etc.

Dans notre contexte de rationalité limitée des acteurs sociaux, où les acteurs disposent d'une vue partielle de l'organisation, l'application de l'apprentissage par logique inductive est impossible. Un acteur, qui a effectué une action positive sur les états de ses relations et a observé une augmentation de sa satisfaction, ne peut pas généraliser en disant que « pour chaque état x d'une relation, il existe une action a positive permettant d'améliorer sa satisfaction » : l'effet d'une action sur la satisfaction d'un acteur ne dépend pas que de l'action effectuée par l'acteur, mais aussi de la réaction des autres acteurs. Et même si elle dépendait uniquement de cette action, l'acteur, disposant d'une vue partielle sur l'organisation, plus précisément sur les fonctions d'effet des relations dont il dépend, ne peut pas connaître l'impact exact de cette action à tout moment.

3.1.3 – L'apprentissage par imitation

L'apprentissage par imitation est une forme d'apprentissage supervisée qui permet à un acteur d'apprendre à effectuer des actions en imitant le comportement des autres acteurs dans son environnement. L'imitation désigne la correspondance entre le comportement de deux acteurs (l'imitateur et l'imité) lorsque cette correspondance résulte de l'observation du comportement du second par le premier. Il s'agit d'un mode d'apprentissage très développé chez les êtres humains, en particulier chez les nouveaux-nés, qui permet d'acquérir un certain nombre d'informations en imitant le comportement des « experts » : c'est par l'imitation que se font tous les apprentissages spontanés de la petite enfance (paroles, gestes, mimique, …) [Piaget, 1962], que de bonnes solutions ont été trouvées pour les problèmes d'évacuation [Faris, 1926], ou encore que des robots ont appris à imiter les comportements d'autres robots [Moga *et al.*, 1998] et [Bakker *et al.*, 1996]. Ce type d'apprentissage nécessite des objectifs et des codes communs de perception et de production chez l'imitateur et l'imité : par exemple, un joueur de rugby ne peut pas imiter le comportement d'un joueur de tennis, un robot de ménage ne peut pas imiter le comportement d'un autopilote.

Cette technique a été récemment utilisée dans les travaux concernant l'apprentissage de robots à agir en observant les comportements des autres robots. [Moga *et al.*, 1998] propose une architecture neuronale permettant à un robot (imitateur) d'apprendre une séquence de mouvements exécutés par un autre robot (imité). [Bakker *et al.*, 1996] propose une approche avec laquelle un robot apprend de nouveaux comportements en observant les comportements des autres robots par imitation.

Dans la modélisation de la sociologie de l'action organisée, une population homogène peut être agrégée en un acteur unique si chacun de ses membres est dans la même situation de dépendance et de contrôle vis-à-vis de toutes les relations (et ce, avec les mêmes enjeux), si bien qu'ils auront des comportements similaires. Dans ce contexte, l'apprentissage par imitation entre les acteurs n'a pas lieu d'être puisqu'un individu ne peut imiter qu'un autre individu qui lui ressemble, et que de ce fait, dans notre modèle, ils constituent un seul acteur. D'une certaine façon, le regroupement d'une population homogène en un seul acteur inclut, par construction, l'apprentissage par imitation : les individus de cette population s'imitent au point d'avoir des comportements similaires.

3.1.4 – L'apprentissage par arbre de décision

L'apprentissage par combinaison de décisions (arbre de décision) est une forme d'apprentissage, fondée sur l'idée simple de réaliser la classification d'un objet par une suite de tests sur les attributs qui le décrivent. Ces tests sont organisés de telle façon que la réponse à l'un d'eux indique à quel prochain test on doit soumettre l'objet en cours d'examen. Le principe de cette règle de décision est d'organiser l'ensemble des tests possibles sous la forme d'un arbre de façon à faire apparaître à l'extrémité de chaque branche les différents résultats possibles en fonction des

décisions prises à chaque étape. En d'autres termes, l'application de cette technique permet d'identifier des sous-espaces de l'espace d'entrée pour lesquels la solution est unique, et lorsqu'un nouveau cas est soumis au système, celui-ci identifie le sous-espace correspondant et retourne la réponse associée. Un exemple très connu est la sélection d'un article pour une conférence scientifique : pour savoir si l'article sera accepté, il devra subir plusieurs tests organisés dans un arbre permettant d'évaluer un article.

[Portet *et al.*, 2008] introduit une méthode d'acquisition automatique des règles de sélection pour l'adaptation en ligne d'un système de monitoring cardiaque. Plus précisément, il s'agit d'une méthode de sélection qui choisit, en ligne, l'algorithme le plus adapté au contexte courant du signal à traiter. Les règles de sélection sont acquises par arbre de décision sur les résultats de performance de 7 algorithmes testés dans 130 contextes différents.

La lisibilité, la rapidité d'exécution, et le peu d'hypothèses nécessaires *a priori* expliquent la popularité des arbres de décision. C'est un outil utilisé dans des domaines variés comme la sécurité, la fouille de données, la médecine, et surtout dans des problèmes relatifs à la classification des données.

Par contre, son application dans le contexte des SAC supposerait que chaque acteur analyse sa situation et le comportement des autres acteurs d'une façon cognitive qui demande beaucoup de connaissances. En d'autres termes, l'acteur doit connaitre les liens de contrôle entre les acteurs et les relations du système, et adapter son comportement en fonction de la distribution de ses enjeux sur les relations d'une part, et d'autre part de la réaction des acteurs qui contrôlent les relations dont il dépend. Cette hypothèse n'est pas vraisemblable dans un contexte de rationalité limitée, les acteurs sociaux disposant d'une vue partielle sur l'organisation, et ne connaissant pas d'avance la structure du jeu.

3.1.5 – Les approches bayésiennes

Les approches bayésiennes sont reconnues comme les solutions les plus efficaces pour résoudre les problèmes contenant de l'incertitude sur les événements qui se produisent dans l'environnement considéré. Considérons l'exemple d'une personne, ayant les yeux fermés, qui va choisir une boite au hasard parmi deux boîtes différentes dont la première, A, contient 90 boules rouges et 10 boules blanches, tandis que la deuxième, B, contient 50 rouges et 50 blanches. Puis elle va tirer toujours au hasard, une boule depuis la boite choisie. Cette boule se trouve rouge. Intuitivement, on se doute que la boîte A a plus de chances d'avoir été choisie. L'approche bayésienne permet de donner une réponse exacte de la probabilité de chaque boîte sachant que la boule est rouge. Notons BA la proposition « la boule vient de la boîte A », BB la proposition « la boule vient de la boîte B », et R l'événement « la boule est rouge ». Connaissant le contenu des boîtes, nous savons que P(R|BA) = 90/100 = 0,9 et P(R|BB) = 50/100 = 0,5, mais ce qui nous intéresse est la probabilité de la boîte A sachant que la boule est rouge. Le théorème de Bayes nous donne :

$$P(BA|R) = (P(BA)^{3} * P(R|BA)) / (P(BA)*P(R|BA + P(BB)*P(R|BB)$$

$$= (0,5 * 0,9) / (0,5 * 0,9 + 0,5 * 0,5) = 0,64$$

Ces approches permettent donc d'exprimer des relations probabilistes entre des ensembles d'évènements. Ces relations diffèrent des relations logiques en ce qu'elles n'autorisent pas un raisonnement implicatif, mais conditionnel : deux évènements peuvent en effet être en relation de co-occurrence sans que l'un implique l'autre. Les approches bayésiennes décomposent le processus d'apprentissage en deux étapes. D'abord, une étape d'inférence de modèles statistiques des données

[3] Si, lorsqu'on a les yeux bandés, les boîtes ne se distinguent que par leur nom, nous avons P(BA) = P(BB) = 0,5, et la somme fait 1.

de type P(y|x), puis une étape de décision s'appuyant sur ces probabilités pour calculer la décision optimale selon l'équation de la règle de décision optimale de Bayes :

$$P(A_i \mid B) = \frac{P(B \mid A_i) \times P(A_i)}{\sum_j P(B \mid A_j) \times P(A_j)}$$ où $\{A_i\}$ est une partition de l'ensemble des possibles.

L'ensemble des faits et des probabilités conditionnelles d'un système de raisonnement peut s'organiser en graphe, sous certaines conditions d'indépendance probabiliste. On peut alors calculer la probabilité conditionnelle de n'importe quel ensemble d'évènements connaissant la probabilité de n'importe quel autre ensemble d'évènements.

Le domaine d'application des approches bayésiennes est très large. [Barrat et al. 2010] utilise cette approche pour définir une méthode pour représenter et classifier des images partiellement annotées (une image est considérée comme partiellement annotée si elle ne possède pas le nombre maximale de mots-clés) sous la forme d'un modèle graphique probabiliste. Cette méthode permet de classer de façon visio-textuelle des images en utilisant l'information graphique et l'information textuelle, et d'étendre automatiquement les annotations existantes à de nouvelles images en prenant en compte les éventuelles relations sémantiques entre mots-clés. [Le Hy *et al.*, 2004] propose une méthode de programmation des comportements basée sur la programmation bayésienne des robots avec un formalisme de description probabiliste. Cette méthode est utilisée par des personnages autonomes, appelées bots, évoluant dans des mondes virtuels, par exemple le jeu vidéo World of Warcraft. Elle permet à un bot d'observer et de copier le comportement d'un humain qui joue à de tels jeux vidéo.

L'emploi des approches bayésiennes dans le contexte du comportement des acteurs sociaux permettrait de bénéficier de leurs avantages concernant l'incertitude des informations collectées par un acteur sur le système. On peut considérer des probabilités de transitions d'un état à un autre état du système (la probabilité de rencontrer l'état E' sachant qu'on était dans l'état E en appliquant une action locale A). Ces probabilités peuvent être initialisées arbitrairement, de façon équiprobable, au début de la simulation, puis modifiées au fur et à mesure en fonction de l'expérience de l'acteur. L'acteur peut alors appliquer la formule de Bayes pour choisir une action lui permettant de se retrouver dans un état qui le satisfait. Par contre, l'emploi de ces approches ne permettrait pas de mettre en relief le processus d'exploration et d'exploitation (cf. 3.2.3). D'autre part, l'emploi des probabilités supposerait que la structure de l'organisation ne change pas. Si l'environnement change, toutes les probabilités changent aussi et l'application de ces probabilités devient donc inefficace.

3.2 – L'approche retenue : l'apprentissage par renforcement

3.2.1 – Présentation

Quand un enfant regarde autour de lui, entend des bruits, sent ou reconnait différentes odeurs (surtout celle de sa mère), joue avec ses bras, et ramasse des objets pour les goûter ou les jeter ailleurs, il ne suit pas des règles explicites. Il réalise une activité spontanée qui, grâce à ses connexions sensorimotrices, lui permet d'interagir avec son environnement et lui fournit une mine de renseignements sur les causes et les effets, sur les conséquences de ses actions, et sur ce qu'il faut faire pour atteindre des objectifs [Piaget, 1962]. Tout au long de nos vies, ces interactions sont une source très importante d'apprentissage sur l'environnement et sur nous-mêmes : apprendre à faire nos premiers pas, à tenir une conversation avec les autres, à conduire une voiture, et à inventer un produit chimique et/ou un algorithme d'intelligence artificielle : nous observons la réaction de l'environnement et nous cherchons à améliorer nos actions afin d'atteindre nos objectifs. L'*apprentissage par renforcement*, nommé aussi *apprentissage par interaction*, est un processus fondamental qui sous-tend presque toutes les théories de l'apprentissage.

L'apprentissage par renforcement est un terme emprunté par [Minsky, 1954] et [Minsky, 1961] à la littérature sur l'apprentissage des animaux. Il englobe une collection variée d'idées ayant des racines dans l'apprentissage des animaux [Barto, 1985] et [Sutton *et al.*, 1987], la théorie du contrôle [Bertsekas, 1989] et [Kumar, 1985], et l'intelligence artificielle [Dean *et al.*, 1991]. Les algorithmes d'apprentissage par renforcement différé ont été utilisés pour la première fois par [Samuel, 1959] et [Samuel, 1967] dans son célèbre ouvrage sur les jeux de dames, puis par [Barto *et al.*, 1983]. L'algorithme Q-Learning de [Watkins, 1989] a constitué une étape importante de la recherche dans ce domaine.

L'apprentissage par renforcement est une approche computationnelle de l'apprentissage par interaction. Il fait référence à une classe de problèmes d'apprentissage dont le but est d'apprendre, à partir d'expériences répétées, un comportement (appelé stratégie ou politique) qui permet de déterminer l'action à exécuter en fonction de l'état courant de façon à optimiser une récompense numérique au cours du temps. L'acteur n'est pas directement informé des actions qu'il peut entreprendre, comme dans la plupart des approches de l'apprentissage supervisé, mais doit découvrir quelle(s) action(s) lui procure(nt) la meilleure récompense. Deux types d'apprentissage par renforcement peuvent être distingués en fonction du domaine d'application : *immédiat*, pour lequel une action n'a d'effets que sur la récompense immédiate, et *différé*, pour lequel une action peut avoir des effets sur le futur (les prochaines situations, et ainsi les prochaines récompenses). Ce dernier type, plus difficile à mettre en œuvre, se rencontre souvent dans le contrôle optimal des systèmes dynamiques et les problèmes de planification de l'intelligence artificielle. Ces deux propriétés - recherche par essai-erreur et récompense différée – sont les deux caractéristiques distinctives les plus importantes de l'apprentissage par renforcement.

En plus de l'acteur et de l'environnement avec lequel il interagit, on peut identifier quatre éléments principaux d'un système d'apprentissage par renforcement :

- *La politique* : est une fonction associant à l'état courant l'action à exécuter. Elle définit la manière de se comporter de l'acteur à un instant donné. L'objectif du processus d'apprentissage est de trouver une politique qui assure une bonne récompense, la plus proche de la politique optimale.
- *La fonction de récompense* : est une fonction qui associe une quantité à chaque état (ou paire état-action) de l'environnement : une récompense. Elle spécifie ce qui est bon immédiatement.
- *La fonction de valeur* (« value fonction » en anglais) : est une fonction qui spécifie ce qui est bon sur le long terme. Elle caractérise le but du problème d'apprentissage.
- *Le modèle de l'environnement* : est un modèle interne de l'environnement tel qu'il est vu par l'acteur. Il lui permet, à partir d'un état et d'une action donnés, de prédire l'état résultant et la récompense correspondante. Cet élément n'est pas essentiel comme les autres, il est utilisé surtout dans les systèmes d'apprentissage exigeant la planification.

3.2.2 – Pertinence de l'apprentissage par renforcement dans notre contexte

L'apprentissage par renforcement présente plusieurs intérêts dans le contexte de la Sociologie de l'Action Organisée :

- Il est le seul procédé disponible quand le comportement désiré est inconnu, soit parce qu'il est impossible de prédire la réaction de l'environnement (dans notre cas, le comportement des autres acteurs), soit parce que cette réaction est trop complexe à modéliser (dans notre cas, cette réaction dépend de plusieurs paramètres, comme les enjeux et les fonctions des autres acteurs, que l'acteur ne connaît pas). Par exemple : un adversaire durant une partie d'échec, le vent durant le pilotage d'un avion, ou les mouvements des adversaires durant un match de foot. Dans certains cas, le comportement désiré est connu mais très complexe à programmer, comme la marche.

- Il prend en considération la non-stationnarité de l'environnement (i.e. le fait que la réaction de l'environnement change avec le temps) : il permet à l'acteur d'adapter constamment son comportement à l'évolution de l'environnement. C'est le cas en ce qui nous concerne, puisque la satisfaction qu'un acteur reçoit à la suite de son action dépend en fait essentiellement de l'action de tous les autres acteurs, et surtout de leurs états respectifs.
- Le comportement « optimal » (celui qui maximise la récompense) n'est pas toujours le « meilleur ». Si l'environnement est peu sûr, le comportement « prudent » (celui qui minimise les risques compte tenu des incertitudes de l'environnement) peut être préférable. Par exemple, on peut concevoir un système de navigation qui prend en considération, d'une part, les distances des itinéraires possibles entre deux endroits différentes, et d'autre part, la probabilité des embouteillages sur ces différents itinéraires. Par conséquence, ce système de navigation sera capable de retrouver le meilleur chemin en fonction des incertitudes de l'environnement. L'apprentissage par renforcement est capable de trouver les deux types de comportement car il prend en considération la non stationnarité de l'environnement.

3.2.3 – Les défis de cette approche

Dans l'apprentissage par renforcement, c'est l'acteur qui produit lui-même les « exemples » à partir desquels il va produire la connaissance qu'il vise à acquérir. Cette approche présente donc une difficulté que l'on ne rencontre pas dans les autres approches d'apprentissage, à savoir le compromis entre l'*exploration* (acquisition de nouvelles connaissances en testant l'effet d'actions très différentes de celles qu'il a entreprises jusqu'alors) et l'*exploitation* (utilisation des connaissances apprises). En effet, l'acteur doit exploiter les connaissances qu'il a acquises de façon à appliquer les actions qui ont prouvé leur efficacité par les récompenses qu'elles ont entraînées, mais il doit aussi explorer les réactions de l'environnement, de façon à en acquérir une bonne connaissance, aussi complète et précise que possible. Ce défi a été mis en évidence par [March, 1991] : si l'exploration est excessive, les connaissances apprises sont sous-utilisées, et si l'exploitation est excessive, peu de connaissances sont acquises. Le dilemme réside dans le fait que l'exploration et l'exploitation sont des dispositions d'apprentissage antinomiques et ne peuvent être poursuivies simultanément.

Un autre défi de l'apprentissage par renforcement est la bonne spécification du but final de l'acteur, la nature de ce qu'il doit apprendre, c'est-à-dire la définition de la fonction de valeur. La clé de l'apprentissage par renforcement est de considérer le problème dans son ensemble, par contraste avec d'autres méthodes qui découpent le problème à résoudre en sous-problèmes, en perdant la vision globale. Par exemple, le but final d'un joueur d'échecs est de prendre le roi de son adversaire, et pas de diminuer le nombre de ses pièces.

3.2.4 – L'interaction Acteur-Environnement

Formalisation

Le modèle standard d'un processus d'apprentissage par renforcement est composé d'un environnement susceptible de se trouver dans un ensemble d'états S, d'un ensemble d'actions A que l'acteur peut réaliser, et d'un ensemble de signaux de renforcements R que l'environnement peut renvoyer à l'acteur. Une fonction d'observation O, pourra être associée à l'acteur, qui détermine comment l'acteur perçoit l'état de l'environnement. Un acteur interagit avec à son environnement par l'intermédiaire des perceptions et des actions. A chaque instant t, il perçoit l'état s_t de l'environnement (ou une observation o_t dans le cas des environnements partiellement observables), il choisit une action a_t, et l'applique sur l'environnement. L'environnement passe à un état s_{t+1}, et retourne un signal de renforcement r_{t+1} à l'acteur. L'acteur met à jour son système d'apprentissage en fonction du signal de renforcement retourné et de son état/observation à l'instant t.

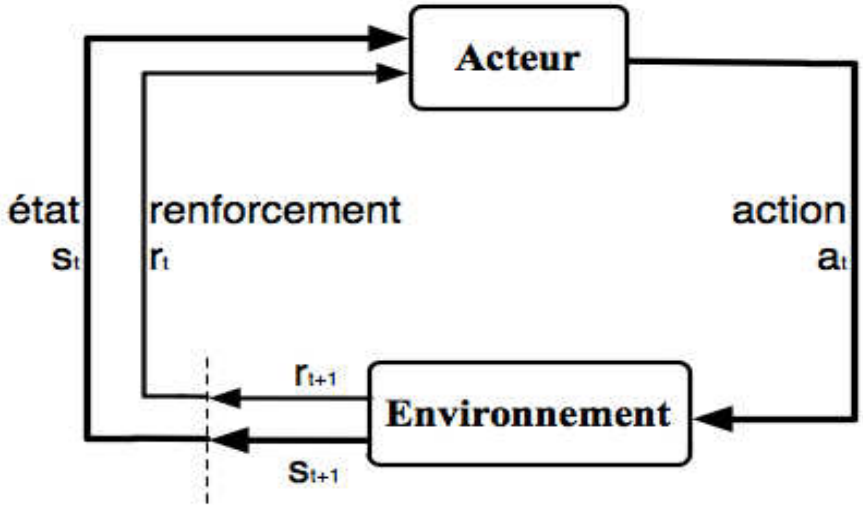


Figure 3.1. *L'interaction acteur-environnement dans l'apprentissage par renforcement.*

Le comportement de l'environnement peut être régi par un Processus de Décision Markovien (PDM, en anglais « Markov Decision Process », MDP) qui décrit l'évolution de l'état de l'environnement et de la récompense en fonction des actions de l'acteur. Cette évolution peut être régie par un Processus de Décision Markovien Partiellement Observable (POMDP) si l'environnement n'est pas complètement observable par l'acteur.

Un Processus Décisionnel de Markov (MDP) est un modèle stochastique issu de la théorie de la décision et de la théorie des probabilités [Bellman, 1957]. Il peut être vu comme un processus stochastique possédant la propriété de Markov[4] auquel on ajoute une composante décisionnelle. Il est utilisé pour étudier un large éventail de problèmes d'optimisation résolus par la programmation dynamique et l'apprentissage par renforcement. Il permet de prendre des décisions lorsque l'on a une certitude sur l'état dans lequel l'acteur se trouve, même en présence d'incertitude sur l'effet des actions.

Formellement, un MDP est un quadruplet {S, A, T, R} où :

- $S = \{s_0, \ldots, s_{|S|}\}$ est l'ensemble fini discret des *états* possibles de l'environnement.
- $A = \{a_0, \ldots a_{|A|}\}$ est l'ensemble fini discret des *actions* que l'acteur peut réaliser.
- $T : S \times A \times S \rightarrow [0 ;1]$ est la *fonction de transition* du système en réaction aux actions de l'acteur : $T(s, a, s') = P\{s_{t+1} = s' / a_t = a ; s_t = s\}$ est la probabilité pour l'environnement de passer de l'état s à l'état s' lorsque l'acteur effectue l'action a. Dans le cas déterministe, la fonction de transition est sous la forme $S \times A \rightarrow S$.
- $R : S \times A \times S \rightarrow \Re$ est la *fonction de renforcement*. Elle retourne la récompense de l'action a effectuée depuis l'état s lorsque l'environnement passe à l'état s'.

Le comportement de l'acteur est défini par une politique $\pi : S \times A \rightarrow [0 ; 1]$ qui guide l'acteur de manière probabiliste en spécifiant pour chaque état s la probabilité de réaliser l'action a, avec :

$$\sum_a \pi(s,a) = 1 \qquad \text{(éq. 3.1)}$$

Le problème à résoudre est donc de permettre à l'acteur de trouver une politique optimale π^* qui maximise l'espérance à l'instant t de sa *Récompense à Long Terme*, RLT_t. RLT_t est définie de

[4] « De manière simplifiée, la prédiction du futur, sachant le présent, n'est pas rendue plus précise par des éléments d'information supplémentaires concernant le passé ; toute information utile pour la prédiction du futur est contenue dans l'état présent du processus ».

différentes manières en fonction de la tâche considérée. Si la tâche consiste à répéter des épisodes qui durent un nombre d'étapes n fixe, RLT_t pourra être la somme des récompenses instantanées pendant un épisode.

$$RLT_t = \sum_{k=0}^{n} r_{t+k+1} \qquad \text{(éq. 3.2)}$$

Si au contraire la tâche se déroule de manière continue sans limite de temps, RLT_t pourra se définir comme la somme des récompenses futures pondérées par une exponentielle décroissante :

$$RLT_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \qquad \text{(éq. 3.3)}$$

où γ est un facteur d'escompte strictement inférieur à 1 indiquant l'importance relative que l'acteur accorde aux récompenses futures.

Les algorithmes d'apprentissage par renforcement que nous verrons plus loin utilisent quasiment tous une *fonction de valeur* V^π qui caractérise la qualité d'une politique π. Cette fonction permet d'estimer la récompense à long terme pour un état donné si l'acteur suit la politique considérée :

$$V^\pi(s) = E_\pi(RLT_t / s_t = s) \qquad \text{(éq. 3.4)}$$

Où E_π est l'espérance pour la politique π de la récompense à long terme. L'évaluation d'une politique peut aussi se définir non pas pour un état mais pour une paire état/action, notée alors Q^π, de la façon suivante:

$$Q^\pi(s, a) = E_\pi(RLT_t / s_t = s, a_t = a) \qquad \text{(éq. 3.5)}$$

Par suite, la fonction de valeur V est alors définie par la moyenne des fonctions de valeur Q pondérées par la probabilité de chaque action :

$$V^\pi(s) = \sum_a \pi(s,a) * Q^\pi(s,a) \qquad \text{(éq. 3.6)}$$

Les fonctions de valeurs permettent de comparer les politiques en définissant un ordre partiel :

$$\pi \geq \pi' \Leftrightarrow \forall\, s,\ V^\pi(s) \geq V^{\pi'}(s) \qquad \text{(éq. 3.7)}$$

$$\pi \geq \pi' \Leftrightarrow \forall\, s, \forall\, a\ Q^\pi(s, a) \geq Q^{\pi'}(s, a) \qquad \text{(éq. 3.8)}$$

Cette relation d'ordre permet de définir la fonction de valeur V* de la politique optimale que l'apprentissage par renforcement cherche à découvrir :

$$V^*(s) = \max_\pi V^\pi(s) \qquad \text{(éq. 3.9)}$$

$$Q^*(s, a) = \max_\pi Q^\pi(s, a) \qquad \text{(éq. 3.10)}$$

Une propriété essentielle des fonctions de valeurs permet de créer les différents algorithmes d'apprentissage : il s'agit de la relation de récurrence connue sous le nom d'*équation de Bellman* :

$$V^\pi(s) = \sum_a \pi(s,a) \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^\pi(s')] \qquad \text{(éq. 3.11)}$$

Où T est la fonction de transition, et R est la fonction de renforcement. Cette équation traduit la sémantique de la fonction de valeur en reliant la valeur à long terme d'un état à la valeur de tous les états qui peuvent lui succéder. Elle se déduit simplement de la définition de V ; une explication détaillée se trouve dans [ENSTA]. Il est également possible d'écrire une relation de récurrence pour

la fonction de valeur de la politique optimale, appelée fonction de valeur optimale. On parle alors d'*équation d'optimalité de Bellman*, qui peut s'écrire :

$$V^*(s) = \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^{\pi}(s')] \qquad \text{(éq. 3.12)}$$

soit, pour la fonction Q :

$$Q^*(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma \max_{a'} Q^*(s',a')] \qquad \text{(éq. 3.13)}$$

Intuitivement, cette équation traduit le fait que la politique optimale choisit l'action qui va maximiser la récompense sur le long terme.

Généralisation

Le cadre présenté ci-dessus est abstrait, flexible. Il peut être appliqué à de nombreux problèmes de différentes façons. De plus, l'environnement peut ne pas respecter plusieurs des propriétés que nous avons implicitement supposées satisfaites :

- Discret : l'ensemble des états et celui des actions sont discrets.
- Déterministe : La fonction de transition du système est déterministe ; T : S × A → S.
- Stationnaire : Dans le cas où la fonction de transition est probabiliste, la valeur de la fonction de transition T(s, a, s') (quelque soit s, a, et s') ne change pas au fil du temps.
- La propriété de Markov : La prédiction du futur, sachant le présent, n'est pas rendue plus précise par des éléments d'information concernant le passé ; toute information utile pour la prédiction du futur est contenue dans l'état présent du processus.

Par ailleurs, on peut considérer que les étapes d'interaction ne produisent pas des intervalles fixes de temps réel, mais font plutôt référence à des étapes successives dans un processus incluant des prises de décision sur l'action. Les actions peuvent être de bas niveau de contrôle, tels que les tensions appliquées aux moteurs d'un bras de robot, ou d'un haut niveau de décision, comme d'aller à un emplacement déterminé. De même, les états peuvent représentés une grande variété d'information, de bas ou de haut niveau.

3.3 – Les différentes méthodes d'apprentissage par renforcement

Les méthodes d'apprentissage par renforcement peuvent être réparties en trois catégories :

- Les méthodes à base de modèle : ce sont des méthodes « hors ligne » ou *a priori* qui exigent la connaissance d'un modèle de l'environnement (les fonctions de transitions et de récompense) complet et parfait de l'environnement.
- Les méthodes sans modèle *a priori* : ce sont des méthodes utilisées dans le cas où l'acteur ne dispose pas d'un modèle de l'environnement, et qui lui permettent d'apprendre ce modèle à partir de ses expériences ; une fois le modèle appris, ces méthodes utilisent les méthodes à base d'un modèle pour calculer un comportement optimal.
- Les méthodes sans modèle : ce sont des méthodes en ligne, utilisées dans les cas où l'acteur ne dispose pas d'un modèle complet de l'environnement, et qui lui permettent de trouver un comportement optimal par l'intermédiaire d'une fonction de valeur optimale, sans pour autant apprendre le modèle de l'environnement.

L'intérêt d'une approche par rapport aux autres dépend beaucoup du problème particulier à résoudre. Les avantages principaux apportés par un modèle de l'environnement sont que

l'expérience réelle peut être complétée par des expériences simulées, et que la connaissance des fonctions de transition et de récompense suffit pour trouver le contrôle optimal. L'inconvénient le plus important est que si l'environnement n'est pas stationnaire, l'acteur doit réapprendre afin de mettre à jour le modèle au fil du temps.

Bien que l'intérêt d'une approche par rapport aux autres ne soit pas complètement établi, il est souvent avantageux d'éviter les méthodes à base de modèle même quand un modèle de l'environnement est disponible. En effet, dans de nombreux problèmes, l'espace d'état est très large, parfois continu et un algorithme à base de modèle fonctionne sur l'intégralité de l'espace ; par contre, un algorithme sans modèle peut se contenter de n'explorer que les parties de l'espace qui sont les plus pertinentes pour le fonctionnement du système.

3.3.1 – Les méthodes à base de modèle

Nous supposons dans cette section que l'acteur apprenant dispose d'un modèle complet et parfait de l'environnement. Nous présentons les méthodes de programmation dynamique qui reposent sur un ensemble d'algorithmes utilisés pour calculer une politique optimale dans un environnement vérifiant les propriétés d'un MDP, en utilisant les équations de Bellman. La programmation dynamique est un paradigme de conception qu'il est possible de voir comme une amélioration ou une adaptation de la méthode « diviser pour régner ». Cette méthode a été introduite dans les années 50 par [Bellman, 1957a] pour résoudre des problèmes d'optimisation. Le terme programmation signifiait à l'époque planification et ordonnancement plutôt que programmation au sens qu'on lui donne de nos jours en informatique. La programmation dynamique n'est pas une méthode d'apprentissage, mais plutôt une méthode computationnelle pour déterminer le comportement optimal étant donné un modèle de la tâche à résoudre. Dans ce cadre, la programmation dynamique va chercher à estimer la fonction de valeur optimale V* afin d'en déduire une politique optimale.

De nombreux travaux sont basés sur la programmation dynamique comme [Berstsekas, 1995], [Bertsekas *et al.*, 1996], [Dreyfus *et al.*, 1977], [Ross, 1983], [White, 1969], [Whittle, 1982, 1983], et [Kumar *et al.*, 1988]. La première connexion entre la programmation dynamique et l'apprentissage par renforcement a été faite par [Minsky, 1961] au sujet des joueurs de dames de [Samuel, 1959]. [Andreae, 1969] a mentionné la programmation dynamique dans un contexte d'apprentissage par renforcement, plus précisément la méthode d'itération de la politique, bien qu'il n'a pas fait des connexions spécifiques entre la programmation dynamique et les algorithmes d'apprentissage. [Werbos, 1977] a suggéré une approche pour approximer la programmation dynamique, appelée « la programmation dynamique heuristique ». [Watkins, 1989] a explicitement lié la programmation dynamique à l'apprentissage par renforcement, en caractérisant une classe de méthodes d'apprentissage par renforcement : « la programmation dynamique incrémentale ».

Dans le reste de cette section, nous présentons deux étapes de la programmation dynamique (évaluation d'une politique et amélioration d'une politique) qui sont utilisées dans deux algorithmes d'apprentissage (itération de la politique et itération de la fonction de valeur). Ensuite nous concluons par une comparaison entre les deux algorithmes introduits.

Les étapes d'évaluation et d'amélioration d'une politique

La première étape de la programmation dynamique est l'estimation de la fonction de valeur d'une politique donnée π. Cela peut se faire en utilisant une procédure itérative qui utilise le fait que la fonction de valeur est le point fixe de l'équation de Bellman (cf. l'équation de Bellman 3.11). On pourra ainsi utiliser cette équation comme étape de mise à jour pour calculer une suite de fonctions (V^k) qui converge vers V^π :

$$V^{k+1}(s) = \sum_a \pi(s,a) \sum_{s'} P(s,a,s')[R(s,a,s') + \gamma V^k(s')] \qquad \text{(éq. 3.14)}$$

Pour fournir une approximation de V^{π}, l'algorithme de calcul va donc itérer jusqu'à ce que les variations $\max_s (|V^{k+1}(s) - V^k(s)|)$ soient inférieures au seuil d'approximation (cf. Algorithme 3.1).

A partir de l'évaluation de la fonction de valeur d'une politique quelconque, il va être possible de calculer une meilleure politique. En effet, pour une politique donnée, il n'y a aucune raison que la fonction de valeur satisfasse l'équation d'optimalité de Bellman (cf. équation 3.12), c'est-à-dire que l'on peut avoir :

$$\pi(s,a) \neq \arg\max_a E(r_{t+1} + \gamma V^{\pi}(s_{t+1}) \mid s_t = s, a_t = a) \qquad \text{(éq. 3.15)}$$

Par contre, on peut toujours prouver que la politique π' définie par :

$$\pi'(s,a) = \arg\max_a E(r_{t+1} + \gamma V^{\pi}(s_{t+1}) \mid s_t = s, a_t = a) \qquad \text{(éq. 3.16)}$$

est meilleure ou équivalente à la politique π, ce qui permet d'améliorer notre politique initiale π. D'ailleurs, si la politique π' ainsi définie n'est pas strictement meilleure que π (c.à.d. $V^{\pi} = V^{\pi'}$), la définition de π' comme solution de l'équation d'optimalité de Bellman prouve donc que la politique π' obtenue est optimale.

L'algorithme d'apprentissage par itération de la politique

L'évaluation et l'amélioration d'une politique peuvent être utilisées de différentes manières pour estimer la politique optimale d'un MDP donné.

Une première méthode est l'itération de la politique. Elle consiste à exécuter les trois étapes suivantes séparément :

- Initialisation
- Évaluation d'une politique
- Amélioration d'une politique

Dans la première étape, la politique et la fonction de valeur sont initialisées arbitrairement. Elles n'ont pas besoin de respecter l'équation de Bellman : la deuxième étape ajustera les valeurs afin qu'elles la respectent. Les étapes deux et trois sont effectuées comme nous venons de le voir.

Ci-dessous, nous présentons un pseudo-algorithme pour la méthode d'itération de la politique, où θ est la précision de l'approximation recherchée de la politique permettant de décider si l'étape de l'évaluation d'une politique est a suffisamment convergée.

θ : le seuil de précision
(S, A, T, R) : un modèle de l'environnement

1. Initialisation
    Choisir une politique arbitraire π, et initialiser sa fonction de valeur V d'une façon arbitraire pour tous les états $s \in S$
Répéter
    2. Évaluation de la politique
    Répéter :
        $\Delta \leftarrow 0$
        Pour chaque $s \in S$:
            $\nu \leftarrow V(s)$
            $V(s) \leftarrow \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V(s')]$
            $\Delta \leftarrow \max (\Delta, |\nu - V(s)|)$
    Jusqu'à $\Delta < \theta$
    3. Amélioration de la politique
    politique_stable ← Vrai
    Pour chaque $s \in S$:
        $b \leftarrow \pi(s)$
        $\pi(s) \leftarrow \arg \max_a \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V(s')]$
        si $b \neq \pi(s)$ alors politique_stable ← Faux
Jusqu'à (politique_stable)

Retourner (π)

Algorithme 3.1. *Pseudo-algorithme de la méthode d'itération de la politique.*

L'algorithme d'apprentissage par itération de la fonction de valeur

Dans la méthode par itération de la politique, l'évaluation et l'amélioration d'une politique sont deux étapes distinctes. L'itération de la fonction de valeur les fusionne. L'amélioration de l'estimation de la fonction de valeur repose toujours sur l'équation de Bellman, mais avec une politique gloutonne. Cette politique choisit l'action qui est estimée la meilleure, donc l'action conduisant au plus grand renforcement estimé. Cette itération est effectuée sur l'ensemble de tous les états. L'itération s'arrêtera si l'amélioration de la fonction de valeur optimale est inférieure à une précision θ. La politique optimale peut être déduite de la fonction de valeur optimale en consultant quelle action retourne le plus grand renforcement sur la partie droite de l'équation de Bellman.

Le Pseudo-algorithme de la méthode par itération de la fonction de valeur est donnée ci-dessous, où $\triangle$ indique la variation maximale de la fonction de valeur après une étape de l'évaluation d'une politique, et θ est la précision de l'approximation recherchée, permettant de décider si l'étape de l'évaluation d'une politique est assez convergente.

θ : le seuil de précision
(S, A, T, R) : un modèle de l'environnement

Initialisation
    Choisir une politique arbitraire π, et initialiser sa fonction de valeur V d'une façon arbitraire pour tous les états $s \in S$
Répéter :
    $\Delta \leftarrow 0$
    Pour chaque $s \in S$:
        $\nu \leftarrow V(s)$
        $V(s) \leftarrow \max_a \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V(s')]$
        $\Delta \leftarrow \max (\Delta, |\nu - V(s)|)$
Jusqu'à $\Delta < \theta$

Pour chaque état $s \in S$:
    $\pi(s) \leftarrow \arg \max_a \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V(s')]$

Retourner (π)

Algorithme 3.2. *Pseudo-algorithme de la méthode par itération de la fonction de valeur.*

Comparaison des deux algorithmes

L'itération de la politique doit parcourir tous les états un plus grand nombre de fois que l'itération de la fonction de valeur, à cause de l'étape de l'évaluation de la politique. Par contre, l'itération de la fonction de valeur doit parcourir l'ensemble des actions possibles dans un état afin de calculer la valeur maximale de la fonction de valeur. Par conséquence, si le ratio état/action est important, l'itération de la politique est plus efficace que l'itération de la fonction de valeur.

D'une façon générale, plus le nombre d'états est grand, plus les calculs prendront du temps. Ainsi, un très grand ensemble d'états est un problème pour chacune des deux méthodes. Cependant, souvent, il n'est pas nécessaire de parcourir l'ensemble de tous les états lors d'une itération, mais uniquement les états les plus pertinents.

3.3.2 – Les méthodes d'apprentissage du modèle

Nous venons de voir les méthodes permettant d'approximer une politique optimale en supposant que l'acteur dispose d'un modèle complet et parfait de l'environnement. Mais il existe des cas où ce modèle n'est pas connu à l'avance par l'acteur, et ce dernier doit découvrir le comportement de son environnement à partir d'interactions.

Dans cette section, nous allons voir des méthodes permettant à un acteur d'interagir avec son environnement afin de construire un modèle de cet environnement. Une fois le modèle construit, l'acteur peut utiliser les méthodes que nous venons de voir pour approximer la politique optimale.

Certainty Equivalent Methods (Méthodes par certitudes équivalentes)

Ces méthodes consistent à apprendre la fonction de transition T(s, a, s') et la fonction de récompense R(s, a) en explorant aléatoirement l'environnement, et en sauvegardant des statistiques sur les résultats de chaque action ; puis à calculer la politique optimale en utilisant une des méthodes à base d'un modèle [Kumar *et al.*, 1986].

Ces méthodes présentent des inconvénients. D'une part, l'exploration aléatoire de l'environnement est problématique : dans la plupart des cas, elle est longue, et s'avère dans certains cas une méthode extrêmement inefficace pour la collecte de données. D'autre part, ces méthodes

font une séparation arbitraire entre la phase d'apprentissage et la phase d'action, ce qui rend problématique la possibilité de changements dans l'environnement : la politique optimale déduite par approximation du modèle appris devient sous optimale ou même invalide si l'environnement change.

La méthode DYNA

Contrairement à Certainty Equivalent Methods, Dyna ne fait pas une séparation arbitraire entre la phase d'apprentissage et la phase d'action : elle permet à l'acteur d'apprendre et d'agir en même temps sur le monde et sur un modèle appris du monde. DYNA est une méthode simple intégrant et faisant un compromis entre trois principes à respecter pour apporter une réponse à la question « comment un acteur décide quoi faire ? ». Le premier est que l'acteur doit déduire sa meilleure action en fonction de son but courant et du modèle de l'environnement. Le deuxième est que l'acteur doit simuler la réaction de l'environnement, et compiler les résultats en un ensemble de réactions, ou de règles situations/actions, qu'il va utiliser pour prendre des décisions en temps réel. Le dernier principe est d'apprendre un ensemble de « bonnes » réactions par essais-erreurs, ce qui présente l'avantage de se dispenser de connaître le modèle de l'environnement à l'avance.

La méthode Dyna, introduite par [Sutton, 1990-1991], est basée sur l'idée que la planification est similaire à un apprentissage par essais-erreurs qui serait mené à partir d'expériences hypothétiques [Craik, 1943] et [Dennett, 1978]. Dyna est basée sur la théorie de la programmation dynamique et sur sa relation avec l'apprentissage par renforcement. Elle utilise les expériences pour construire un modèle (les fonctions T' et R'), et utilise les expériences et le modèle construit pour ajuster la politique. Sutton a introduit deux architectures de Dyna : Dyna-PI, qui est basée sur la programmation dynamique, plus précisément sur la méthode de l'itération de la politique, et peut être liée à des idées de l'intelligence artificielle telles que les fonctions d'évaluation et les plans universels (les systèmes réactifs), et Dyna-Q qui est basée sur le Q-Learning de Watkin. D'après Sutton, l'algorithme de Dyna-PI ne s'adapte pas aux changements dans l'environnement. Par contre, Dyna-Q est facile à adapter pour une utilisation dans des environnements dynamiques. Nous ne détaillerons donc que l'algorithme de Dyna-Q.

Le Pseudo-algorithme de Dyna-Q est donné ci-dessous. Les étapes (a) à (d) sont identiques à celles de l'algorithme de Q-Learning (cf. algorithme 3.7) introduit dans la section suivante. ε-greedy est une politique qui choisit la meilleure action avec une probabilité (1 - ε) et une action au hasard avec une probabilité ε. À l'étape (e), le modèle de l'environnement est mis à jour. L'étape (f) s'effectue hors-ligne. Au lieu d'utiliser un modèle parfait de l'environnement, des informations acquises en-ligne sont utilisées pour apprendre par Q-Learning, mais hors-ligne. Les paires état/action sont choisies au hasard en fonction des états visités précédemment.

S : ensemble des états possibles de l'environnement
A(s) : ensemble des actions possibles pour un état s de l'environnement
α : est le taux d'apprentissage tel que : $0 < \alpha \leq 1$
γ : est un facteur d'escompte tel que : $0 \leq \gamma < 1$
K : est une constante choisie arbitrairement par le modélisateur

Initialisation de Q (s, a), T (s, a), R(s, a) arbitrairement pour tous $s \in S$ et $a \in A(s)$
Répéter toujours :
(a) s ← l'état courant (non terminal)
(b) a ← ε-greedy(s, Q)
(c) exécuter l'action a ; observer l'état résultant, s', et le renforcement, r
(d) $Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q'(s, a)]$
(e) T(s, a) ← s' et R(s, a) ← r (supposant que l'environnement est déterministe)
(f) répéter K fois :
s ← un état observé précédemment au hasard
a ← une action précédemment effectué dans l'état s
s' ← T(s, a) et r ← R(s, a)
$Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q'(s, a)]$

Algorithme 3.3. *Pseudo-algorithme de la méthode Dyna-Q.*

L'étape f de l'algorithme de la méthode Dyna-Q fait en sorte que cet algorithme nécessite K fois le calcul de Q-learning (algorithme 3.7), ce qui est largement inférieur au calcul requis par les méthodes à base d'un modèle. Une valeur raisonnable de K est déterminée en fonction de la vitesse relative du calcul et du délai de la prise de décision.

Bien que la méthode Dyna soit une grande amélioration des méthodes précédentes, elle souffre d'être relativement non orientée : elle met à jour aléatoirement des paires d'état/action, plutôt que de se concentrer sur les états pertinents.

La méthode Prioritized Sweeping (balayage prioritaire)

Les problèmes rencontrés dans la méthode Dyna, notamment la désorientation de la mise à jour aléatoire des paires d'état/action, sont résolus par la méthode de balayage prioritaire [Moore *et al.*, 1993]. L'algorithme du balayage prioritaire est similaire à celui de Dyna, sauf que les mises à jour ne sont plus choisies au hasard. Cette méthode utilise une queue à priorité des paires état/action. La paire état/action de plus grande priorité sera choisie en premier. La priorité de chaque paire dépend de la valeur absolue de l'erreur ($| r + \gamma \max_{a'} Q(s', a') - Q(s, a)|$). Plus la priorité d'une paire est importante, plus la position de cette paire avance dans la queue.

Le pseudo-algorithme de la méthode du balayage prioritaire est donné ci-dessous. L'étape (e) calcule l'erreur qui détermine la priorité de la paire état/action. Si cette erreur est plus grande qu'une certaine valeur, la paire sera ajoutée à la queue de priorité à l'étape (f). Dans le cas contraire, cette paire ne sera pas évaluée hors-ligne.

S : ensemble des états possibles de l'environnement
A(s) : ensemble des actions possibles pour un état s de l'environnement
α : est le taux d'apprentissage tel que : $0 < \alpha \leq 1$
γ : est un facteur d'escompte tel que : $0 \leq \gamma < 1$
K : est une constante choisie arbitrairement par le modélisateur
θ : est le seuil de l'erreur

Initialisation de Q (s, a), T(s, a), et R(s, a) arbitrairement pour tous $s \in S$ et $a \in A(s)$, PQueue est vide
Répéter toujours :
(a) s ← l'état courant (non terminal)
(b) a ← politique (s, Q) (par exemple ε-greedy)
(c) exécuter l'action a ; observer l'état résultant, s', et le renforcement, r
(d) T(s, a) ← s' et R(s, a) ← r
(e) p ← $| r + \gamma \max_{a'} Q(s', a') - Q(s, a)|$
(f) si $p > \theta$, alors insérer (s, a) dans PQueue avec une priorité p
(g) répéter K fois, tant que PQueue n'est pas nul :
(s, a) ← Premier (PQueue) (retourne la paire avec la plus grande priorité)
s' ← T(s, a) et r ← R(s, a)
$Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q'(s, a)]$
Répéter, pour tous les prédécesseurs s'', a'' l'action qui a conduit à s :
r'' ← le renforcement prédit
p ← $| r'' + \gamma \max_{a} Q(s, a) - Q(s'', a'')|$
si $p > \theta$, alors insérer s'', a'' dans PQueue avec une priorité p

Algorithme 3.4. *Pseudo-algorithme de la méthode du balayage prioritaire.*

L'avantage principal de cet algorithme est que si un acteur rencontre une variation importante de Q(s, a), il la propage vers les états prédécesseurs pertinents. Le balayage prioritaire est plus performant que la méthode Dyna-Q car il se concentre sur les parties « intéressantes » de l'espace d'états.

3.3.3 – Les méthodes sans modèle

Dans cette section, nous supposons comme dans la section précédente que l'acteur ne dispose pas d'un modèle de l'environnement et qu'il doit interagir avec son environnement afin d'approximer la politique optimale, mais ici sans pour autant apprendre le modèle.

La question principale posée dans cette section est de permettre à l'acteur de savoir si une action effectuée était bonne ou mauvaise. Une première stratégie, dite Monte-Carlo, est d'attendre la « fin » de la tâche, et de récompenser l'action si l'acteur a bien accompli son but ou la punir dans le cas contraire. Cependant, il peut être difficile de savoir quand une tâche est accomplie, et surtout il se peut que ce soit alors trop tard. De plus, cela pourrait nécessiter une très grande quantité de mémoire. Une autre stratégie, dite différence temporelle, est de considérer le renforcement immédiat et la valeur estimée de l'état suivant. La suite de cette section présente des méthodes qui utilisent ces deux stratégies.

La méthode Monte Carlo

La dénomination de la méthode de Monte-Carlo a été introduite en 1947 par Nicholas Metropolis, et publiée pour la première fois dans [Metropolis *et al.*, 1949]. En 1998, Sutton a présenté la méthode Monte-Carlo dans [Sutton *et al.*, 1998]. Cette méthode utilise les mêmes idées que la programmation dynamique (estimer la fonction de valeur puis améliorer la politique), mais

en ayant recours à des expériences réalisées dans l'environnement plutôt qu'à un modèle ; L'exigence d'un modèle complet de l'environnement *a priori* pour approximer la politique optimale rend les méthodes de programmation dynamique peu utiles. L'estimation de la fonction de valeur V se fait à partir d'un ensemble de séquences, appelée épisode, « état-action-récompenses-état-action-…-étatTerminal» réalisées par l'acteur. Après avoir généré un nombre M d'épisodes en utilisant la politique π, Il est alors possible d'estimer la fonction de valeur de cette politique pour un état s par la moyenne des renforcements :

$$V^{\pi}(s) = \frac{1}{N(s)} \sum_{épisode_i \,/\, i=1}^{M} R_i(s) \qquad \text{(éq. 3.17)}$$

Où N(s) est le nombre d'épisodes dans lesquels apparaît s, et $R_i(s)$ est le renforcement après la première visite (first-visit)[5] de l'état s dans un épisode $épisode_i$ (si s n'apparaît pas dans $épisode_i$, alors $R_i(s) = 0$). Il est également possible d'intégrer les nouvelles séquences de manière itérative en utilisant une mise à jour du type :

$$V(s) \leftarrow V(s) + \alpha[R(s) - V(s)] \qquad \text{(éq. 3.18)}$$

Où α est un taux d'apprentissage compris entre 0 et 1.

Cette méthode s'applique de la même façon pour une fonction Q(s, a), ce qui est encore plus intéressant, car pour trouver la politique optimale à partir de V*, il faut disposer d'un modèle de l'environnement, ce qui n'est pas nécessaire en utilisant Q*. L'utilisation de Q et de la méthode de Monte-Carlo permet donc de découvrir la politique optimale sans aucun modèle de l'environnement, en utilisant uniquement des expériences réalisées dans cet environnement.

La méthode de Monte-Carlo doit estimer les valeurs Q(s, a) à partir des récompenses obtenues après avoir réalisé l'action a dans l'état s. Ceci suppose donc que tous ces couples (s, a) soient rencontrés. Il faut donc ajouter au comportement défini par la politique un comportement d'exploration qui va assurer que toutes les actions seront réalisées avec une certaine probabilité (même faible). Deux solutions existent pour résoudre ce problème. La première consiste à contraindre les politiques pour qu'elles associent à toutes les actions au moins une faible probabilité proportionnelle à un paramètre ε. L'apprentissage converge alors vers la politique optimale au sein de cette classe. Cette méthode s'appelle contrôle « on Policy » car elle modifie la politique effectivement suivie par l'acteur et évalue cette politique modifiée. L'autre méthode est une méthode « *off Policy* » car elle évalue une politique tandis que l'acteur en suit une autre. Cette autre politique choisit en général l'action de la politique originale avec une probabilité 1 - ε et une action aléatoire avec une probabilité ε. Pour évaluer la politique originale, l'évaluation de Monte-Carlo utilise simplement les parties finales des épisodes pour lesquelles le choix d'action correspond au choix qui aurait été fait par la politique originale, mais modifie les pondérations des récompenses pour compenser les différences de probabilité de choix des actions entre les deux politiques. La méthode « *on Policy* » est plus rapide que la méthode « off-policy », car l'algorithme « off-policy » ne fait pas la mise à jour de toutes les paires état-action rencontrées dans un épisode. Donc il nécessite plus d'épisodes pour trouver la politique optimale. Le pseudo-script de l'algorithme « *on Policy* » de Monte Carlo est présenté ci-dessous :

[5] Il est également possible que $R_i(s)$ soit la moyenne des renforcements après chaque visite de l'état s (every-visit) dans $épisode_i$

S : ensemble des états possibles de l'environnement
A(s) : ensemble des actions possibles pour un état s de l'environnement
k(s) : le nombre d'épisodes dans lesquels apparaît s

Initialisation : Pour tous $s \in S$ et $a \in A(s)$:
  $Q(s, a) \leftarrow$ arbitrairement
  $\pi(s, a) \leftarrow$ une politique ε-greedy arbitraire
  $R(s) \leftarrow 0$
  $k(s) \leftarrow 1$
Répéter toujours :
  (a) Réaliser un épisode en utilisant π
  (b) Pour chaque paire (s, a) dans cet épisode :
    $R(s) \leftarrow$ renforcement rencontré dans la première visite de (s, a)
    $Q(s,a) \leftarrow Q(s,a)\frac{k}{1+k} + R(s)\frac{1}{1+k}$
    $k(s) \leftarrow k(s) + 1$
  (c) Pour chaque s dans cet épisode :
    $a^* = \text{argmax}_a\, Q(s, a)$
    Pour chaque $a \in A(s)$

$$\pi(s, a) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|A(s)| & \text{si } a = a^* \\ \varepsilon/|A(s)| & \text{si } a \neq a^* \end{cases}$$

Algorithme 3.5. *Pseudo-algorithme de la méthode Monte-Carlo.*

La convergence presque sûre de cet algorithme vers la fonction V est assurée sous des hypothèses générales [Bertsekas *et al.*, 1996].

L'apport principal de la méthode Monte Carlo réside donc dans la technique qui permet d'estimer la valeur d'un état sur la base du cumul des différentes récompenses associées à cet état lors d'épisodes distincts. Un autre avantage est qu'il est possible de se concentrer sur un sous-espace d'états (les états visités durant un épisode), les autres étant négligés. La contrepartie de cet avantage est que la convergence soit très lente car cette méthode nécessite un très grand nombre d'épisodes afin de trouver la meilleure solution.

Il est toutefois possible d'améliorer cet algorithme en autorisant la mise à jour de la fonction de valeur non plus à la fin de chaque trajectoire, mais à la suite de chaque transition du système. Nous développons ici cette variante, dont le principe est à la base des méthodes de différence temporelle.

Les méthodes de différence temporelle

L'idée principale des méthodes à différence temporelle est dérivée de la différence entre les estimations successives temporelles de la même quantité, par exemple la probabilité de gagner à un jeu de Tic-Tac-Toe. [Samuel, 1959] est le premier à proposer et mettre en œuvre une méthode d'apprentissage qui comprenait les idées de différence temporelle, dans le cadre de son célèbre jeu de dames. Ces méthodes combinent deux techniques. D'une part l'incrémentalité de la programmation dynamique : la valeur estimée de $V(s_t)$ est mise à jour en fonction de la valeur estimée de $V(s_{t+1})$; il y a donc une propagation de la valeur estimée de l'état courant en fonction des valeurs estimées des états successeurs. Et d'autre part le recours à l'expérience des méthodes de Monte-Carlo : chacune de ces valeurs résulte d'une estimation locale qui repose sur l'expérience accumulée par l'acteur au fil de ses interactions avec son environnement.

Dans ce paragraphe, deux méthodes à différence temporelle sont présentées : sarsa-learning (méthode « on-policy »), et Q-Learning (méthode « off-policy »). Les performances de ces deux

méthodes sont égales dans la plupart des cas. Cependant, Q-Learning converge vers la solution optimale, tandis que Sarsa converge vers la solution « prudente » (cf. 3.2.2).

Une méthode de différence temporelle : Sarsa-Learning

SARSA-Learning est un algorithme d'apprentissage d'une politique dans un processus décisionnel Markovien [Sutton *et al.*, 1998]. Le nom SARSA a d'abord été exploré par [Rummery *et al.*, 1994] qui l'a nommé Q-Learning modifié (en anglais, « modified Q-Learning »), puis introduit par [Sutton, 1996]. Le nom de SARSA reflète simplement le fait que la fonction principale de mise à jour des valeurs de Q dépend de l'état actuel de l'acteur (s_t), l'action choisie par l'acteur (a_t), le renforcement reçu par l'acteur en effectuant cette action (r_t), l'état suivant selon la fonction de transition (s_{t+1}), et l'action que l'acteur va choisir dans ce nouvel état (a_{t+1}). D'où le quintuplet (s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}) utilisé pour réaliser la mise à jour de la valeur d'une paire état/action à l'étape t sous la forme suivante :

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \, [r_{t+1} + \gamma \, Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \qquad \text{(éq. 3.19)}$$

Cette mise à jour est en fait une combinaison de l'équation de Bellman et de l'équation différentielle moyenne ($Q_{new} = Q_{old} + \alpha \, (X - Q_{old})$). Effectuer ces mises à jour implique que l'acteur détermine par anticipation quelle est l'action a_{t+1} qu'il réalisera lors de l'étape suivante, lorsque l'action a_t dans l'état s_t l'aura conduit dans l'état s_{t+1}. Il en résulte une dépendance étroite entre la question de l'apprentissage et la question de la détermination de la politique optimale que l'acteur réalise conjointement l'apprentissage d'une bonne politique et l'interaction opérationnel avec l'environnement. Dans un tel cadre, il n'existe qu'une seule politique qui doit prendre en compte à la fois les préoccupations d'exploration et d'exploitation et l'acteur est contraint de réaliser cet apprentissage uniquement sur la base de la politique qu'il suit effectivement. On dit d'un algorithme tel que SARSA qu'il est on-policy.

Un script de l'algorithme de sarsa est présenté ci-dessous :

```
Initialisation : Q(s, a) arbitrairement
Répéter (pour chaque épisode, ou toujours) :
    Initialiser s
    Choisir l'action a dans s selon une politique dérivée de Q (par exemple :ε-greedy)
    Répéter
        Réaliser l'action a, observer r et s'
        Choisir l'action a' dans s' selon une politique dérivée de Q (par exemple :ε-greedy)
        Q(s, a) ← Q(s, a) + α [r + γ Q(s', a') - Q(s, a)]
        s ← s'
        a ← a'
    Jusqu'à s est un état terminal (l'objectif est atteint)
```

Algorithme 3.6. *Pseudo-algorithme de la méthode de SARSA.*

Une deuxième méthode de différence temporelle : Q-Learning

L'algorithme Q-learning, publié dans [Watkins *et al.*, 1992], se présente comme une simplification de l'algorithme SARSA par le fait qu'il n'est plus nécessaire pour l'appliquer de déterminer un pas de temps à l'avance quelle sera l'action réalisée au pas de temps suivant. Son équation de mise à jour est la suivante :

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \, [\, r_{t+1} + \gamma \, \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]^6 \qquad \text{(éq. 3.20)}$$

[6] Cette formule peut s'écrire sous la forme : $Q(s_t, a_t) \leftarrow (1 - \alpha) \, Q(s_t, a_t) + \alpha \, [\, r_{t+1} + \gamma \, \max_a Q(s_{t+1}, a)]$

La différence essentielle entre SARSA et Q-learning se situe au niveau du terme d'erreur. Le terme $Q(s_{t+1}, a_{t+1})$ apparaissant dans l'équation de mise à jour de la valeur d'une paire état/action de SARSA a été remplacé par le terme $\max_a Q(s_{t+1},a)$ dans l'équation de Q-Learning. Cela pourrait sembler équivalent si la politique suivie était *greedy*[7] (on aurait alors $a_{t+1}= \arg\max_a Q(s_{t+1}, a)$). Toutefois, compte tenu de la nécessité de réaliser un compromis entre exploration et exploitation, ce n'est généralement pas le cas. Il apparaît donc que l'algorithme SARSA effectue les mises à jour en fonction des actions choisies effectivement alors que l'algorithme Q-learning effectue les mises à jour en fonction des actions optimales, même si ce ne sont pas ces actions optimales que l'acteur réalise, ce qui est plus simple.

Un script de l'algorithme de Q-Learning est présenté ci-dessous :

```
Initialisation : Q(s, a) arbitrairement
Répéter (pour chaque épisode, ou toujours) :
    Initialiser s
    Répéter
        Choisir une action a selon une politique dérivée de Q (par exemple :ε-greedy)
        Réaliser l'action a, observer r et s'
        Q(s, a) ← Q(s, a) + α [r + γ max_a' Q(s', a') - Q(s, a)]
        s ← s'
    Jusqu'à s est un état terminal (l'objectif est atteint)
```

Algorithme 3.7. *Pseudo-algorithme de la méthode de Q-Learning.*

La méthode de différence temporelle à n-étapes

L'objectif de cette méthode est d'associer les méthodes Monte Carlo et à différences temporelles, plus précisément de faire un compromis entre ces deux approches. Ces deux méthodes enregistrent les états visités pour permettre à l'acteur d'apprendre à partir de ses expériences. La différence entre les méthodes Monte Carlo et à différences temporelles est la longueur de la séquence utilisée pour l'apprentissage (un épisode vs un seul état). Il est possible de définir une méthode intermédiaire : la différence temporelle à n-étapes. Cette méthode consiste à faire une sauvegarde pour un certain nombre d'étapes n d'étapes au lieu d'une seule étape comme pour la différence temporelle. Le diagramme de la figure 3.2 montre l'idée centrale de la différence temporelle à n-étapes : si n = 1 (respectivement n = max), on retrouve la méthode de différence temporelle (respectivement Monte Carlo).

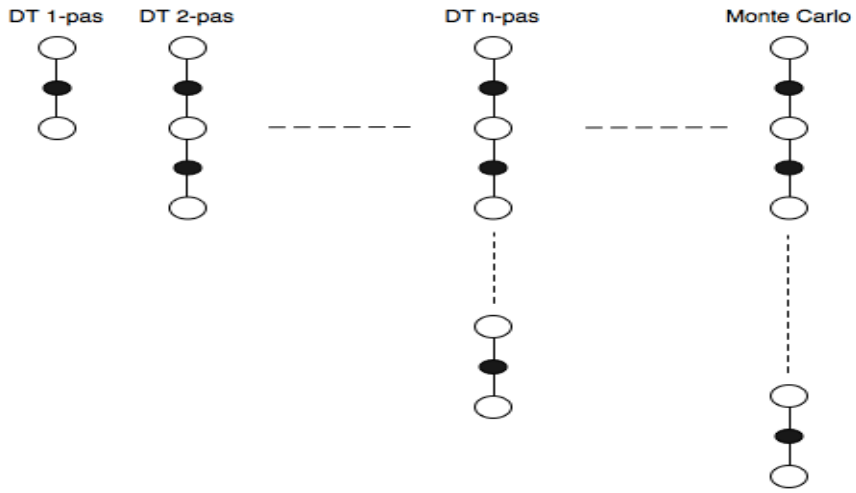


Figure 3.2. *La longueur de la séquence utilisée pour l'apprentissage [Dolk, 2010].*

[7] La politique greedy consiste à appliquer toujours la meilleure action.

Cette méthode utilise un paramètre λ-return, d'où la notation DT(λ). Ce paramètre sera utilisé pour calculer le renforcement d'un état pour n étapes de la façon suivante :

$$R_t^{\lambda} = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}R_t^{(n)} \qquad \text{(éq. 3.21)}$$

L'implémentation de cette équation dans les algorithmes de Q-Learning et Sarsa n'est pas possible car elle nécessite l'utilisation des renforcement futurs qui sont inconnus. Une solution est d'utiliser le passé (backward view) au lieu d'essayer de prédire le futur (forward view). Le chapitre 7.4 du [Sutton *et al.*, 1998] montre que les deux points de vue sont équivalents.

Cette technique forme un pont entre la méthode de Monte Carlo et la différence temporelle. Son utilisation donne lieu à un algorithme d'apprentissage plus rapide car elle utilise les estimations d'une manière plus efficace. L'inconvénient principal est de ne pas savoir *a priori* la valeur optimale du paramètre n, qui ne peut être trouvée qu'empiriquement.

3.4 – L'apprentissage par renforcement multi-Agent (ARMA)

3.4.1 – Présentation des SMA

D'après [Weiss, 1999] et [Vlassis, 2003], un système multi-agent (SMA) est un groupe d'entités autonomes, appelées aussi agents ou acteurs, en interaction, partageant le même environnement qu'ils perçoivent par des capteurs, et sur lequel ils interagissent par des actionneurs. Les systèmes multi-agents ont de nombreux domaines d'application telles que le contrôle distribué, les équipes de robots, la gestion des ressources, etc. Ils peuvent constituer le moyen le plus naturel de décrire un système, ou peuvent offrir une perspective distribuée sur des systèmes qui sont à l'origine considérés comme centralisés [Busoniu *et al.* 2008].

La généralisation du processus de décision de Markov aux systèmes multi-agent est *le jeu stochastique* défini comme un n-uplet $\{S, A_1, \ldots, A_n, T, r_1, \ldots, r_n\}$ où n est le nombre des acteurs, S est l'ensemble discret des états de l'environnement, A_i, i = 1, …, n sont les ensembles discrets des actions pour les acteurs, T est la fonction de transition du système, et r_i, i = 1, …, n sont les fonctions de renforcements des acteurs.

Les jeux stochastiques peuvent être classés en trois catégories en fonction des buts des acteurs [Busoniu *et al.,* 2008] :

- Coopératif : un jeu stochastique est coopératif si $r_1 = \ldots = r_n$. Tous les acteurs ont le même but, et cherche donc à maximiser le même renforcement. Par exemple, une équipe de football où tous les joueurs d'une même équipe ont exactement le même but commun : gagner le match.
- Compétitif : un jeu stochastique est compétitif si $r_1 + \ldots + r_n = 0$. Les acteurs ont des buts opposés et chacun cherche à réaliser le sien tout en empêchant les autres de réaliser les leurs. Par exemple, dans un jeu de Tennis, les deux joueurs sont engagés dans un jeu compétitif ; chacun a intérêt à gagner le match, ce qui implique que l'autre perde.
- Mixte : un jeu stochastique est mixte si il n'est ni coopératif ni compétitif.

Les agents dans un SMA peuvent être programmés *a priori* pour avoir des comportements déterminés à l'avance, mais il est souvent nécessaire qu'ils soient capables d'apprendre en temps réel, afin d'adapter leurs comportements aux réactions des autres et aux changements de l'environnement. Dans certain cas, les agent doivent être capable de créer des nouveaux comportements au fur et à mesure afin d'améliorer leur performance ou celle du système multi-agent. L'apprentissage par renforcement, qui permet à un acteur d'apprendre à partir de ses expériences, peut être appliqué dans les SMA : sa simplicité et sa généralité rendent son application intéressante surtout quand l'acteur ne dispose pas *a priori* d'un modèle de l'environnement.

3.4.2 – Avantages vs Défis

En dehors de l'intérêt intrinsèque de l'apprentissage par renforcement, que nous avons présenté précédemment, plusieurs avantages spécifiques tiennent à la nature même des systèmes multi-agents. Un de ces intérêts est l'accélération de l'apprentissage rendu possible par le calcul parallèle quand les acteurs exploitent la structure décentralisée de la tâche à résoudre [Guestrin *et al.*, 2002] [Busoniu *et al.*, 2005]. De plus, dans le cas de tâches semblables et de même objectif à résoudre par plusieurs acteurs, ils peuvent bénéficier du partage de l'expérience, soit par la communication entre ces acteurs [Tan, 1993], soit par l'imitation du comportement [Clouse, 1995] et [Price *et al.*, 2003]. Un autre avantage est la possibilité de remplacer un acteur par un autre quand ce premier tombe en panne : par exemple, dans un jeu de football où un joueur peut en remplacer un autre. En effet, la plupart des SMA sont conçus pour permettre l'insertion rapide et facile de nouveaux acteurs dans le système en temps réel. L'apprentissage est donc une technique adéquate pour la mise en œuvre de l'ouverture, la flexibilité, l'adaptabilité ou l'évolutivité, propriétés essentielles des SMA.

L'apprentissage par renforcement multi-agent (ARMA) pose de nouveaux défis, en plus de ceux rencontrés dans l'apprentissage par renforcement pour un seul acteur que nous avons présentés dans la section 3.2.3. La spécification de l'objectif de l'apprentissage reste toujours une des grandes difficultés dans le contexte multi-agent. D'une part, dans un jeu stochastique général, les renforcements reçus par les acteurs sont corrélés et il est impossible de les maximiser indépendamment les uns des autres. D'autre part, Les objectifs de l'apprentissage évoqués dans la littérature intègrent à la fois la stabilité et l'adaptation du comportement de chaque acteur. La stabilité, qui signifie essentiellement la convergence de l'apprentissage vers une politique stationnaire, est nécessaire car les comportements stables des acteurs sont le plus souvent indispensables au bon fonctionnement du système. Cependant l'adaptation assure que la performance est maintenue, voire améliorée, face aux changements de politiques des autres acteurs ou à l'évolution de l'environnement. Par conséquent, une bonne définition de l'objectif doit inclure ces deux composantes.

Une des difficultés est la non stationnarité de l'apprentissage dans un contexte multi-agent. Cette difficulté découle du fait que tous les acteurs apprennent simultanément, et que chacun se trouve face à un problème d'apprentissage non-stationnaire : la meilleure politique d'un acteur change suite aux changements des comportements des autres acteurs. Un autre défi est la dimensionnalité qui tient à la croissance exponentielle de l'espace des paires état/action avec l'augmentation du nombre d'états du système et d'actions possibles pour l'acteur. Cette complexité est exponentielle en le nombre d'acteurs, car chacun doit ajouter et calculer ses propres variables. Un dernier défi est la spécification d'un but global de l'apprentissage par renforcement multi-agents. En effet, les renforcements reçus par les acteurs sont, en général, corrélés et il est impossible de les maximiser simultanément. Le problème est alors de déterminer la grandeur, s'il en existe une, qui doit être optimisée par l'ensemble du système.

3.4.3 – Les algorithmes d'ARMA

Dans cette section, nous présentons brièvement les algorithmes d'ARMA groupés selon le type du jeu (coopératif, compétitif, et mixte), en nous inspirant très largement de [Busoniu *et al.*, 2008].

Les jeux coopératifs

Dans les jeux coopératifs, les acteurs sont indépendants les uns des autres, mais ils cherchent à réaliser un objectif commun. Considérons l'exemple d'une équipe de foot constituée de plusieurs joueurs où chacun est autonome, agit d'une façon indépendante des autres acteurs, et cherche à gagner le jeu en agissant au mieux en fonction de sa position : soit en défendant, soit en attaquant, soit en faisant le lien entre la défense et l'attaque. La coordination d'actions entre ces acteurs

permet de minimiser les efforts inutiles, de maximiser le renforcement, et d'atteindre le but général commun à tous les acteurs : battre l'adversaire et gagner le match.

Les algorithmes de cette section peuvent être classés en trois catégories selon le degré de la coordination entre les acteurs.

Dans la première catégorie, on suppose que les acteurs ont la même fonction de récompense R qu'ils essayent de la maximiser. Cela fait qu'il existe une seule fonction Q à apprendre par les acteurs, nommée Q-Learning d'équipe (Team Q-Learning) par [Littman, 2001], et que la coordination entre les acteurs est donc inutile.

La deuxième catégorie est constituée de méthodes basées sur le suivi des autres acteurs permettant ainsi la coordination indirecte entre les acteurs. Ces méthodes consistent à adopter des « bonnes » actions qui sont évaluées grâce aux modèles des autres acteurs estimés par l'acteur lui-même [Claus *et al.*, 1998], ou des statistiques des valeurs observées dans le passé (cf. 5.2.2, l'algorithme de l'objectif global).

La troisième catégorie repose sur la coordination directe entre les acteurs à l'aide de conventions sociales, de rôles, et de communications. Les conventions sociales et les rôles restreignent l'ensemble des actions parmi lesquelles l'acteur doit choisir. Les conventions sociales encodent des préférences *a priori* à l'égard des actions conjointes [Boutilier, 1996], tandis qu'un rôle restreint l'ensemble des actions possibles [Spaan *et al.*, 2002]. Les communications sont utilisées pour négocier les choix d'actions [Fischer *et al.*, 2004]. Les communications sont utilisées pour négocier les choix d'actions [Fischer *et al.*, 2004]. Par ailleurs, [Matignon *et al.*, 2012] limite la communication entre les acteurs (des robots) par l'utilisation d'une fonction de valeur distribuée. Adoptée par une équipe des robots cherchant à couvrir un certain environnement en temps réel, cette fonction permet à chacun de calculer localement une stratégie qui minimise les interactions entre les robots et maximise la couverture de l'espace par l'équipe. De même, [Guestrin *et al.*, 2002] simplifie la coordination quand la fonction globale à optimiser peut être décomposée de manière additive en plusieurs fonctions locales qui ne dépendent que des actions locales de chaque acteur.

Les jeux compétitifs

Dans les jeux compétitifs, les acteurs ont des buts opposés, et chacun cherche à réaliser son objectif quelque soit l'action des autres acteurs (et à empêcher les autres de réaliser leurs objectifs). Le principe de minimax s'impose étant le meilleur principe dans des tels jeux. Ce principe consiste à évaluer toutes les actions possibles en fonction du renforcement apporté à l'acteur lui-même (et aux adversaires lorsque les renforcements sont publics), et à choisir l'action qui maximise son renforcement (tout en minimisant ceux des adversaires). L'algorithme minimax-Q [Littman, 1994 et 2001] utilise le principe du minimax pour calculer les stratégies et les valeurs dans les jeux répétitifs, et une règle de différence temporelle similaire à Q-Learning pour propager les valeurs à travers les paires états-actions.

Les jeux mixtes

Dans les jeux mixtes, on ne peut faire aucune hypothèse sur les propriétés des fonctions de renforcement des acteurs. Chacun a son propre objectif qui peut être soit commun ou soit en conflit avec un ou plusieurs autres acteurs. Le concept d'équilibre de la théorie des jeux est essentiel dans les algorithmes pour les jeux mixtes. Lorsque plusieurs équilibres existent dans un jeu particulier, le problème de la compatibilité des équilibres recherchés par chacun des acteurs se pose, et chaque acteur participe plus ou moins en fonction de ses capacités à la sélection de l'équilibre.

En plus des méthodes basées sur le suivi des autres acteurs, les algorithmes présentés précédemment dans le cas d'un seul acteur (comme Q-Learning, SARSA, etc.) peuvent être appliqués dans les systèmes multi-agents. Ces méthodes ne font aucune hypothèse sur le type du jeu, et sont donc applicables sur les jeux mixtes. Cependant, aucune convergence de ces méthodes

n'est garantie ; les acteurs agissent et apprennent simultanément, ce qui rend le problème d'apprentissage non-stationnaire.

Les méthodes basées sur le suivi des autres acteurs permettent à un acteur d'estimer des modèles de stratégies ou de politiques des autres, et d'agir en utilisant la meilleure action (réponse) à ces modèles. Ces méthodes supposent donc que chaque acteur est capable d'observer les acteurs du système, leurs situations et leurs actions afin de construire un modèle interne pour chaque acteur (ou un modèle global du système). Étant donné que tous les acteurs agissent et apprennent en même temps, ces méthodes visent donc la meilleure réponse au changement de l'environnement (aux actions des autres acteurs) [Brown, 1951].

En outre, ils existent d'autres méthodes basées sur le suivi des autres acteurs mais recherchent plutôt la convergence, ainsi que l'adaptation de la stratégie de chacun à celles des autres acteurs. Ces méthodes supposent que chaque acteur est capable d'observer le comportement des autres, mais aussi qu'il dispose d'un modèle complet et parfait de son environnement lui permettant de calculer analytiquement tous les éventuels équilibres du système. [Conitzer *et al.*, 2003] propose un algorithme AWESOME, adapter lorsque tout le monde est stationnaire, sinon passer à l'équilibre (en anglais, « Adapt When Everyone is Stationary, Otherwise Move to Equilibrium »), permettant à un acteur de changer son comportement en fonction de ceux des autres ; il observe les autres acteurs et, quand il détecte la non-stationnarité de leur comportement, il change le sien en passant de la meilleure réponse dans le jeu stochastique à un comportement lui permettant d'atteindre à un équilibre de Nash précalculé.

3.4.4 – Les domaines d'application de l'ARMA

L'apprentissage par renforcement multi-agent a été appliqué à une grande variété de domaines. Le domaine d'application le plus connu pour les systèmes multi-agent coopératifs est le contrôle distribué où un ensemble d'acteurs autonomes agissent en parallèle pour résoudre une tâche commune : le contrôle de processus [Stephan *et al.*, 2000], le contrôle de feux de circulation [Wiering, 2000], [Bakker *et al.*, 2005], ou le contrôle de réseaux électriques [Riedmiller *et al.*, 2000]. L'apprentissage par renforcement multi-agent a été souvent utilisé dans le domaine des équipes des robots : l'observation multi-cible est une extension de la tâche d'exploration, où les robots doivent maintenir un groupe de cibles mouvantes à l'intérieur d'un certain périmètre [Touzet, 2000], [Fernandez *et al.*, 2001]. La poursuite implique la capture des cibles mouvantes par une équipe de robot [Kok *et al.*, 2005], [Ishiwaka *et al.*, 2003]. L'apprentissage par renforcement a été aussi appliqué par des acteurs négociateurs qui agissent sur les marchés électroniques au nom d'une société ou d'une personne, en utilisant des mécanismes tels que les négociations et les ventes aux enchères [Wellman *et al.*, 2003]. Dans la gestion des ressources, les acteurs forment une équipe coopérative constituée de contrôleurs qui chacun contrôle une seule ressources et apprend à gérer les requêtes de service en optimisant la performance [Crites *et al.*, 1998], et de clients qui doivent apprendre la façon optimale de sélectionner des ressources [Schaerf *et al.*, 1995].

3.5 – Le jeu social et l'ARMA

Dans le contexte de rationalité limitée de la modélisation des organisations dans SocLab, les acteurs sociaux disposant d'une vue partielle sur l'organisation ne connaissent pas la structure du jeu, et ils explorent leur environnement sans disposer d'informations sur les autres acteurs. Ils ne savent ni combien d'acteurs sont présents, ni leurs objectifs, et ne sont donc pas en mesure d'observer directement leur comportement afin de s'en construire un modèle interne. En d'autres termes, chaque acteur voit le jeu comme s'il n'existait qu'un seul autre acteur (l'environnement) avec lequel il va négocier son comportement : tenter de coopérer si la coopération est rentable, et ne pas coopérer dans le cas contraire.

Bien que le jeu social relève de l'apprentissage multi-agent par renforcement, les méthodes disponibles dans la littérature, et dont quelques unes sont présentées dans les sections précédentes de ce chapitre, ne sont pas applicables pour modéliser le processus de régulation des organisations sociales par lequel chaque acteur découvre une façon de se comporter qui soit acceptable par tous, y compris lui-même. Les spécificités du jeu social tiennent à la structure du jeu, aux règles de ce jeu, la façon dont les acteurs y jouent, et à l'objectif du jeu, les résultats que l'on en attend.

Concernant la *structure* du jeu, elle peut être caractérisée de la façon suivante[8] :

1. Le jeu est déterministe, la fonction de transition peut donc être définie comme $T : S \times A \rightarrow S$.
2. Le jeu n'est pas coopératif, chaque acteur ayant une fonction de renforcement qui lui est propre, et il n'est pas non plus à somme nulle.
3. La contribution aux fonctions de renforcement de l'action de chaque acteur est indépendante de l'action des autres acteurs. Introduisons quelques notations pour formaliser cette propriété. Soit $N = \{1, \ldots, n\}$ l'ensemble des acteurs. Notons R_i la fonction de renforcement de l'acteur i, a_i une action choisie par l'acteur i, $u = (a_1, \ldots, a_n) = (a_i)_{i \in N}$ une action conjointe de tous les acteurs, et, pour $J \subset N$ un sous-ensemble d'acteurs, $u_J = (a_i)_{i \in J}$ un vecteur d'actions des acteurs appartenant à J et $u_{-J} = (a_i)_{i \in N \setminus J}$ un vecteur d'actions des acteurs n'appartenant pas à J et, par commodité, $u_{-i} = u_{-\{i\}}$. Le jeu social vérifie alors $\forall\, i, j \in N, \forall\, s \in S, \forall\, u_{-i}, u'_{-i}, a_i, a'_i$

   $$R_j (s, (u_{-i}, a_i)) - R_j (s, (u_{-i}, a'_i)) = R_j (s, (u'_{-i}, a_i)) - R_j (s, (u'_{-i}, a'_i))$$

   puisque, en notant $s = (s_r)_{r \in R}$ et $a_i = (d_r)_{r \in R \text{ contrôlée par } i}$, on a :

   $R_j (s, (u_{-i}, a_i)) - (R_j (s, (u_{-i}, a'_i)) = \sum_{r \in R \text{ contrôlée par } i} \text{enjeu}(j, r) * (\text{effet}_r(j, s_r+d_r) - \text{effet}_r(j, s_r+d'_r))$, grandeur qui ne dépend que des actions réalisées par l'acteur i.

   Cette propriété assure que, par son action, chaque acteur exerce une influence spécifique sur les fonctions de renforcement, qui peut donc, même de façon très bruitée, être perçue comme telle par tous les acteurs. Elle pourrait être formalisée plus simplement en prenant en compte le fait que les renforcements ne dépendent que de l'état du jeu et pas des actions, mais cette propriété ne semble pas essentiel dans les résultats de l'algorithme présenté au chapitre suivant.

La principale *règle du jeu* sociale est que chacun y joue indépendamment des autres, ne disposant pas d'information sur ce qu'il en est des autres, et qu'il ne connaît pas la structure ni l'état courant du jeu :

1. Les joueurs ne communiquent pas entre eux et, à chaque étape du jeu, ils ne connaissent ni l'action réalisée ni le renforcement obtenu par les autres.
2. Les joueurs ne connaissent pas l'état du jeu, ils ne le perçoivent qu'indirectement par le renforcement qu'ils obtiennent. Les joueurs ne connaissent pas non plus la structure du jeu, sa fonction de transition, mais cela ne leur serait d'aucune utilité puisqu'ils ne connaissent pas les actions réalisées par les autres joueurs.

Ces deux règles font qu'un joueur ne dispose pas des informations lui permettant de se construire un modèle des autres acteurs ou du jeu dans son ensemble ou de calculer ce que serait les

[8] Pour axiomatiser la structure des jeux sociaux, il faudrait vraisemblablement prendre en compte une quatrième propriété, à savoir que les contributions de chaque acteur sont agrégées additivement dans les fonctions de renforcement.

équilibres du jeu. Nous verrons, au chapitre 5, que la règle (1) doit être assouplie si l'on veut que les acteurs ne soient pas purement individualistes.

Enfin, la principale singularité du jeu social tient à *l'objectif* du jeu, défini comme le résultat d'une partie ou bien comme la solution que l'on souhaite obtenir en résultat d'un algorithme exécuté par les joueurs d'un jeu social particulier. Il ne s'agit pas du tout d'une stratégie pour chacun des joueurs, qui lui indique l'action qu'il doit réaliser dans chaque état du jeu afin que la somme escomptée de ses renforcements soit maximale, ce qui constitue le critère d'évaluation de la qualité de ce résultat.

1. Le jeu social ne produit un résultat que s'il parvient à un état stationnaire.
2. Le résultat d'une partie d'un jeu social est l'état stationnaire auquel le jeu est parvenu ou, de façon équivalente, le renforcement perçu par chacun des acteurs dans cet état. La qualité de ce résultat est évaluée par la pertinence de la distribution des renforcements entre les acteurs en fonction de la structure du jeu.

La nature même des résultats attendus d'un jeu social fait que les équations de Bellman ne sont pas applicables. Et, de toute façon, tel qu'il est vu par chaque acteur, le jeu social est trop instable pour qu'il puisse être considéré comme markovien, même de façon (très) partiellement observable. A la différences de l'orientation générale des algorithmes relevant de l'ARMA, un modèle d'apprentissage pour les partenaires d'un jeu social ne cherche pas à ce que chaque acteur apprenne une « meilleure réponse » au comportement des autres. ([Soham et al., 2007] fait d'ailleurs remarquer que, dans un jeu multi-agent, chacun des partenaires est autant enseignant qu'apprenant : chacune de ses actions tient compte des actions passées des autres, mais aussi influence leurs actions à venir. Il conviendrait donc de parler « d'apprentissage interactif » ou « d'adaptation multi-agent » plutôt que « d'apprentissage multi-agent »). Si cette meilleure réponse est appliquée par chacun, elle constitue un équilibre de Nash et correspond donc à un état du jeu dans lequel chacun se prémunit contre le pire cas. A l'inverse, le processus d'apprentissage doit faire en sorte que chaque acteur prenne le risque de coopérer et, dans la mesure de ses moyens, contraigne les autres à faire de même. Les états recherchés ne sont pas *a priori* des états d'équilibre, au sens où nul n'aurait intérêt à être le seul à s'en éloigner ; ce n'en sont pas moins des états relativement stables, dans la mesure où celui qui s'en écarte est rapidement sanctionné par les autres. Pour que les acteurs trouvent comment ils peuvent coopérer au bénéfice de chacun, le processus d'apprentissage doit donc réserver une large part à *l'exploration* des potentialités du jeu, disposition dont March a souligné l'importance dans tout processus d'apprentissage [March, 1991].

3.6 – La prise de décision en rationalité limitée

Il n'est pas question de faire un état de l'art sur la théorie de la décision, mais d'examiner un certain nombre de modèles algorithmiques de la rationalité limitée qui ont été proposés comme alternatives au modèle de la rationalité parfaite de l'homo economicus, optimisateur bayésien de son utilité subjective espérée, dont de nombreux auteurs ont souligné le manque de validation empirique. On peut distinguer trois principaux types de travaux dans lesquels il est fait référence à la rationalité limitée :

- La réalisation d'une tâche purement cognitive de résolution de problème consistant par exemple à trouver par essais-erreurs l'équation d'une fonction [Edmonds, 1999], à répondre à une question fermée à partir de certaines indications [Gigerenzer *et al.*, 1996], ou à trouver un meilleur chemin dans l'arbre d'un jeu [Jehiel, 1998] ou d'un graphe [Gabaix *et al.*, 2000].
- Dans ces situations, les modèles proposés relèvent marginalement d'une rationalité limitée, dans la mesure où les alternatives et l'évaluation de leur qualité sont définies par la structure même du problème à résoudre. Il serait plus exact de les qualifier d'heuristiques

qui cherchent à approximer le mieux possible une solution connue mais qui ne peut être calculée faute d'informations et/ou de ressources cognitives suffisantes.

- L'étude de l'émergence de propriétés globales dans un contexte d'interaction où les agents sont dotés d'un comportement relativement fruste (voir [Lupi, 1998], [Dal Forno *et al.*, 2002] parmi beaucoup d'autres, par exemple, la revue JASSS).
- Là encore, il s'agit essentiellement d'heuristiques qui ne font pas référence avec précision aux caractéristiques de la rationalité mise en œuvre. La plupart des travaux de ce type sont centrés sur l'émergence et mettent en évidence l'écart entre le comportement relativement simple des individus et les propriétés remarquables qu'elles entraînent, au niveau macro.
- La construction par apprentissage d'une représentation (ou d'un modèle, ou d'extension d'état) du contexte dans lequel l'acteur doit prendre une décision, représentation qui lui permettra de déterminer quelles sont les différentes alternatives qui s'offrent à lui, de les évaluer et finalement de sélectionner celle qui lui assure le meilleur gain [Schuster, 2009] et [Sun *et al.*, 2005].
- Dans ce cas, il s'agit de modèles cognitifs qui demandent de grandes capacités cognitives, et ce de façon explicite, en appliquant des règles sur les conditions et la nature de l'évolution de cette représentation, règles assez sophistiquées dans le cas de ces deux travaux.

Dans ces modèles, il s'agit de permettre à un acteur de choisir la stratégie qui optimise un critère prédéfini (la fonction d'utilité qui évalue le bénéfice qu'il retire de l'effet de sa stratégie sur le contexte de son action), ou du moins de s'en approcher le plus possible. Ces modèles négligent donc l'un des principes essentiels de la rationalité limitée selon [Simon, 1982], à savoir que les acteurs ne cherchent qu'une solution « satisfaisante », la réalisation de leurs « aspirations » qui ne sont pas prédéfinies mais dépendent du contexte[9]. S'agissant de la régulation des systèmes d'action concrets, il est clair que cette dimension ne peut pas être négligée : le niveau de satisfaction dont chaque acteur peut se satisfaire n'est pas prédéfini, puisqu'il doit être accepté par chacun des autres acteurs et est donc le résultat d'une négociation.

Parmi ces modèles, il convient de distinguer celui proposé par [Dutech *et al.*, 2003], qui propose de construire une extension d'état, appelée observable exhaustif, composée d'éléments non ambigus, de dimension finie, appelés trajectoires d'observations/action dans les environnements partiellement observables. Cette extension d'états rend l'environnement complètement (ou presque) observable, ce qui permet alors d'utiliser les techniques d'apprentissage classiques.

Cependant l'algorithme proposé, comme toutes autres méthodes exigeant la construction du modèle par apprentissage, n'est pas capable de prendre en compte la non stationnarité de l'environnement ; en d'autres termes, il n'est pas réactif. Il n'est donc pas très adapté pour la prise de décision par des acteurs autonomes qui doivent réagir en temps réel à des modifications de leur environnement.

Il convient, aussi, de distinguer celui proposé par [Selten, 1998], qui prend en compte les « aspirations » de l'acteur. Ces aspirations sont représentées sous la forme de vecteurs qui associent une valeur cible à chacun des objectifs de l'acteur. Une aspiration est faisable lorsqu'elle est dominée (composante par composante) par le résultat d'au moins l'une des alternatives. L'acteur diminue la valeur cible de l'un de ses objectifs si son aspiration courante n'est pas faisable, sinon il

[9] Notons toutefois que, dans certains des modèles de rationalité limitée concernant les agents d'un jeu économique spatialisé, le niveau d'aspiration d'un agent est la moyenne des gains obtenus par ses voisins, voir par exemple [Dalmagro *et al.,* 2006 ; Lupi, 1998]. Même si ce niveau peut varier, il reste donné et non pas construit par l'acteur, et cela n'est possible que parce que, le jeu étant symétrique, les agents sont homogènes ; ce n'est bien sûr pas le cas des jeux sociaux, dans lesquels le statut de chaque acteur est spécifique.

augmente l'une de ces valeurs. Il est satisfait, et donc arrête le processus de recherche, lorsque son aspiration est faisable et maximale (i.e. elle n'est dominée par aucune autre aspiration faisable). Le réaménagement des préférences de l'acteur est propre à chacune de ses aspirations et donc parfaitement contextualisé.

Ce modèle a le mérite de mener de front la recherche d'un objectif qui soit satisfaisant et la recherche d'une solution qui le rende réalisable. Il y manque cependant un mécanisme essentiel qui déterminerait comment un acteur manipule son aspiration, c'est-à-dire sélectionne l'objectif dont la valeur cible doit être augmentée ou diminuée[10]. De ce fait, il ne peut pas donner lieu à des simulations.

[10] Selten déplace ce problème en introduisant la notion de « schéma d'aspiration » qui associe à chaque aspiration l'objectif dont la valeur cible doit, le cas échéant, être modifiée.

Chapitre 4 – Le comportement des acteurs sociaux

Dans toute organisation sociale, les comportements des acteurs dans les relations qu'ils entretiennent les uns avec les autres sont relativement stabilisés. D'ailleurs, une organisation dans laquelle les comportements de chacun seraient erratiques et imprévisibles les uns vis-à-vis des autres ne saurait réaliser ses objectifs et donc perdurer, ne pouvant pas permettre les anticipations indispensables à la coordination des acteurs. La stabilisation du comportement des acteurs est le résultat d'un processus que la SAO appelle la régulation, au cours duquel chacun ajuste son comportement à celui des autres de façon à se trouver dans une situation qui le satisfasse.

L'étude des configurations d'un SAC nous renseigne sur ce que sa structure permet aux acteurs de faire, elle ne nous renseigne pas sur les comportements qu'ils sont susceptibles d'adopter effectivement ; par exemple, il n'est pas certain qu'un acteur soit disposé à coopérer avec d'autres en l'absence de réciprocité. C'est cette question de la faisabilité sociale des configurations d'une organisation que la simulation nous permet d'aborder : à partir d'un état initial quelconque, nous pouvons doter les acteurs d'une rationalité qui permette à chacun de jouer le jeu social à la recherche d'un comportement qui le satisfasse, et ce jusqu'à ce que le SAC se stabilise dans un état stationnaire dans lequel chacun accepte la situation dans laquelle il se trouve, c'est-à-dire soit régulé.

Dans ce chapitre, nous exposons tout d'abord les principes de la régulation du comportements des acteurs et la façon dont nous les avons modélisés. Puis nous décrivons l'algorithme proprement dit. Ensuite, nous caractérisons les résultats produits par l'algorithme de simulation. Enfin, nous établirons l'importance de chacun des paramètres psycho-cognitives par des analyses de sensibilité.

4.1 – Le modèle de la rationalité des acteurs sociaux

La SAO postule que les acteurs sociaux sont stratégiques, motivés par une visée, un objectif non nécessairement explicite, et qu'il exercent leur stratégie dans le cadre d'une rationalité limitée. En effet, l'acteur ne connaît pas la structure complète de l'organisation dont il fait partie[11] : il perçoit sa situation sous la forme d'un vecteur des impacts des relations dont il dépend et l'évalue sous la forme d'une variable agrégée, appelée *satisfaction* (cf. 4.2.2.Satisfaction).

Dans cette section, nous rappelons brièvement le concept de rationalité limitée, puis nous présentons le processus de décision des acteurs. Nous présentons ensuite une notion qui est fondamentale dans ce processus : l'ambition d'un acteur. Enfin, nous terminons en donnant la boucle principale de l'algorithme de simulation.

4.1.1 – L'hypothèse de la rationalité limitée

La rationalité limitée est un concept introduit en 1955 par Herbert A. Simon [Simon, 1955] et [Simon, 1982]. Il est notamment utilisé en sociologie et en micro-économie. Il porte sur le comportement d'un acteur dans une situation de prise de décision. Contrairement au postulat de rationalité substantive traditionnellement utilisé en économie, il considère que l'acteur a un comportement rationnel, mais que sa rationalité est limitée en termes de capacité cognitive et d'information disponible ; ceci est particulièrement vrai dans le domaine social, du fait de l'opacité de tout système social, du caractère plus ou moins implicite des règles, « toujours ambivalentes » selon [Friedberg, 1993], et des différentes interprétations que chacun peut en faire [Roggero *et al.*, 2003]. Il est donc difficile pour l'acteur d'appréhender complètement les conséquences de ses

[11] Connaître la structure de l'organisation reviendrait à raisonner en information complète et à déterminer *a priori* quel est l'état optimal du système au regard de ses objectifs.

actions. L'acteur ne peut « concevoir qu'un nombre limité de solutions pour résoudre le problème auquel il doit faire face. Le champ des comportements possibles est donc limité et souvent, la décision relève davantage d'une logique stimulus-réponse que de l'arbitrage raisonné entre plusieurs alternatives, basé sur une analyse rationnelle et parfaite » [Scieur, 2005]. Selon ces hypothèses, le comportement d'un acteur ne consiste pas à optimiser son choix vis-à-vis de ses *aspirations*, mais à choisir la première solution trouvée satisfaisante[12] dans sa situation concrète, en évitant de consommer un temps et une énergie excessives à effectuer son choix. La rationalité limitée repose ainsi sur trois notions : l'*imperfection de l'information*, la *difficulté de l'anticipation* et le *nombre limité des comportements envisageables*.

SocLab comporte un algorithme de simulation dont les résultats indiquent comment il est plausible que les acteurs d'une organisation sociale se comportent les uns vis-à-vis des autres. Cet algorithme, qui met en œuvre les principes de l'auto-apprentissage par essais-erreurs et le renforcement des règles apprises, implémente les trois principes susmentionnés en dotant les acteurs d'une vision locale de la structure de l'organisation, d'une ambition évolutive (cf. 4.1.3), et d'un processus de décision (cf. 4.1.2) qui vise une situation satisfaisante et non pas une situation optimale.

4.1.2 – Le processus de décision d'un acteur

Il s'agit de trouver un mécanisme de rationalité limitée pour modéliser la façon dont les acteurs d'un SAC sont susceptibles de déterminer leur comportement.

[Simon, 1982] a fait valoir l'utilisation d'algorithmes plausibles, psychologiquement fondés sur le concept de rationalité limitée. Dans ce cadre, la rationalité de l'acteur suppose qu'il sélectionne une action sur la base de son effet escompté, à savoir faire évoluer le système vers une situation préférable pour lui. Son comportement est ainsi fondé sur un cycle de trois opérations de base :

- *Perception* : collecter des informations sur sa propre situation et sur le contexte.
- *Décision* : trouver des actions envisageables, les évaluer et sélectionner celle(s) qui semble la plus souhaitable dans sa situation actuelle compte tenu de ses objectifs.
- *Action* : exécuter l'action sélectionnée, dans la mesure où elle s'avère effectivement réalisable.

Pour rendre compte de cette procédure, le paradigme de l'apprentissage par essais-erreurs et renforcement [Sutton *et al.*, 1998] présente l'avantage de ne faire aucune hypothèse *a priori* sur la façon dont un acteur social pourrait « calculer » la façon de se comporter. Selon cette approche, l'acteur expérimente les réactions de son environnement (i.e. la variation de sa satisfaction) aux actions qu'il entreprend (i.e. la modification de l'état des relations qu'il contrôle) pour apprendre progressivement le meilleur comportement.

Dans l'implémentation simple de ce paradigme que nous avons adoptée, chaque acteur se construit et gère une base de règles qui lui indiquent ce qu'il peut ou doit faire dans certaines circonstances. Chaque règle est de la forme (situation, action, force) où :

- *Situation* : désigne le domaine de validité de la règle, représentée par la liste des effets sur l'acteur de chacune des relations dont il dépend ; c'est la situation de l'acteur au moment de la création de la règle.
- *Action* : désigne la liste des variations à apporter à l'état de chacune des relations que l'acteur contrôle ; ces variations sont choisies au hasard au moment de la création de la règle.

[12] Simon a utilisé le terme « satisficing » pour dénoter une situation suffisamment satisfaisante, combinaison des termes « satisfy » et « sufficing » [Simon, 1955]

- *Force* : est une évaluation de l'efficacité de la règle, initialisée à 0 et augmentée ou diminuée selon l'influence de l'application de cette règle sur la satisfaction de l'acteur, de la façon suivante :

$$\text{Règle.Force}_t = (1 - \alpha) \times \text{Règle.Force}_{t-1} + \alpha \times \Delta\text{Satisfaction} \qquad \text{(éq. 4.1)}$$

où α est un taux d'apprentissage entre 0 et 1, qui gère l'importance relative entre l'estimation précédente et la nouvelle expérience. Cette mise à jour relève de la technique « Q-Learning » (cf. 3.3.3. Q-Learning). Son atout principal est sa facilité d'implémentation : le cœur de cette méthode n'excède pas quelques lignes de code. De plus, un acteur dispose au début d'une mémoire vide qu'il va faire évoluer au gré de ses expérimentations, aucune connaissance initiale sur la structure de l'organisation et sur la réaction des autres acteurs n'est requise. Bien sûr, il reste un point indispensable : l'acteur doit être capable de déterminer la récompense engendrée par une de ses actions en fonction de l'état dans lequel il se trouve.

4.1.3 – L'ambition d'un acteur

D'après [Molm, 1991], un acteur social typique doit être en mesure de réaliser, même d'une façon approximative, une évaluation de sa situation pour savoir si sa satisfaction courante est plutôt bonne ou plutôt mauvaise par rapport à ce qu'elle pourrait être, et de se comporter d'une façon différente dans l'un ou l'autre cas. Nous avons donc attaché à chaque acteur une variable *ambition* qui représente son objectif ; un acteur s'estimera satisfait si la satisfaction que lui procure l'état courant du jeu est supérieure ou égale à son ambition, et le jeu s'arrête lorsque chacun des acteurs est satisfait : un état régulé de l'organisation a été trouvé puisqu'aucun acteur n'a de raison de chercher à adopter un autre comportement. L'ambition d'un acteur est initialisée à la valeur maximale de satisfaction que la structure du jeu lui accorde et elle évolue en fonction de la situation courante, selon un principe de réalité : elle augmente lorsque la satisfaction de l'acteur est plus grande que son ambition et elle diminue dans le cas contraire.

La solution retenue pour le dilemme exploration/exploitation, présenté dans le paragraphe 3.2.3 du chapitre 3, a un effet directe sur la convergence de l'algorithme : une exploration excessive est en général trop coûteuse, et un état stable sera difficilement atteint, tandis que l'algorithme s'arrêtera à la première solution trouvée, sans en rechercher d'autres peut-être meilleures, si l'exploitation est excessive. Il convient donc d'adapter le taux d'exploration/exploitation à l'étape du processus de recherche, en privilégiant l'exploration au début pour, progressivement, exploiter de plus en plus les connaissances acquises. C'est l'écart entre son ambition et sa satisfaction courante qui permettra à un acteur de régler la valeur de ce paramètre : plus sa satisfaction est en dessous de son ambition, plus il va explorer, alors qu'il va privilégier l'exploitation si sa satisfaction est proche ou supérieure à son ambition. Nous dirons qu'un acteur est « *satisfait* » si sa satisfaction est supérieure ou égale à son ambition.

4.1.4 – La boucle principale de la simulation

L'algorithme de simulation du jeu social est ordonnancé par une boucle principale selon laquelle les acteurs sélectionnent leur action indépendamment les uns des autres.

Si, à une étape de cette boucle, tous les acteurs sont satisfaits, ils n'ont plus besoin de modifier leur comportement : un état stationnaire est atteint, l'organisation est alors régulée. Tant que ce n'est pas le cas, la simulation se poursuit et s'arrête, en tout état de cause, lorsqu'est atteint le nombre maximal de pas que l'expérimentateur aura défini.

A : est l'ensemble des acteurs.
R : est l'ensemble des relations.
Initialisation :
étape ← 0
nbPasMax ← une valeur définie par le modélisateur au début de la simulation
Répéter
Répéter (pour chaque acteur a ∈ A)
$situation_a$ ← perception ( ) ;
$action_a$ ← selection.action ( ) ;
Répéter (pour chaque relation r ∈ R)
appliquer (r, $action_{controler(r)}$ (r))
étape ← étape + 1
Tant que (Il existe un acteur qui n'est pas satisfait) **et** (étape < nbPasMax)

Algorithme 4.1. *Pseudo-code de la boucle principale de l'algorithme de simulation.*

Reste à remarquer que nous dissocions les opérations de base de notre algorithme : la perception, la décision, et l'action. Cela revient à considérer que les acteurs prennent leurs actions simultanément et indépendamment les uns des autres. En d'autres termes, l'action d'un acteur n'influence pas la décision d'un autre acteur à la même étape de simulation.

4.2 – L'algorithme de simulation

L'algorithme présenté ici ne prétend pas décrire comment les acteurs sociaux effectivement choisissent de se comporter au sein d'un système d'action auquel ils participent : nous ne savons pas grand chose à ce sujet et il y a une grande diversité d'individus et de systèmes d'action. Nous préférons dire qu'il présente plutôt une façon dont les acteurs pourraient se déterminer.

Pour tenir compte de la variabilité des acteurs, nous introduisons quatre paramètres psycho-cognitifs auxquels le modélisateur pourra attribuer des valeurs différentes pour chacun des acteurs de l'organisation considérée. Nous introduisons ensuite les principales variables de l'algorithme et explicitons les relations qui existent entre ces variables et les paramètres. Enfin, nous présentons l'algorithme que chacun exécute indépendamment des autres.

4.2.1 – Les paramètres psycho-cognitifs d'un acteur

Les paramètres psycho-cognitifs permettent d'individualiser la façon dont chaque acteur délibère. Les valeurs de ces paramètres sont fixées par le modélisateur selon les hypothèses qu'il veut tester.

La ténacité

La valeur de ce paramètre, dont l'échelle est un entier compris entre 1 et 10, détermine la propension de l'acteur à privilégier l'exploration ou l'exploitation. En effet, la valeur de la ténacité d'un acteur joue un rôle principal dans la mise à jour de son taux d'exploration (cf. figure 4.1) : plus un acteur est tenace, plus son taux d'exploration est important, et plus l'acteur va explorer et tenter d'améliorer sa situation. En conséquence d'une exploration plus intense, la simulation est susceptible de durer longtemps avant qu'un état stationnaire soit atteint.

Une grande valeur (10) correspond à une forte tendance à explorer pour une longue durée et avec une intensité forte, tandis qu'une faible valeur (1) correspond à une forte tendance à exploiter. La valeur 0 est exclue car un acteur qui n'explore pas du tout n'a rien à exploiter.

La réactivité

Ce paramètre, dont l'échelle de valeur est un entier compris entre 1 et 10, détermine l'importance relative que l'acteur accorde au présent et au passé dans son processus d'apprentissage. Il détermine la rapidité avec laquelle l'acteur prend en compte la réaction de l'environnement, notamment la variation de sa satisfaction.

Une grande valeur de la réactivité (10) correspond à un acteur sans mémoire qui actualise son taux d'exploration et son ambition en fonction de sa seule situation courante. Tandis qu'avec une faible valeur (1), l'acteur diffère l'actualisation de son taux d'exploration et l'adaptation de son ambition, et donc attache plus d'importance au poids du passé. La valeur 0 est exclue car un acteur pas du tout réactif fera toujours la même action et n'apprendra rien.

Le discernement

A chaque étape de la simulation, l'acteur évalue la *proximité* entre sa situation courante (la liste des effets de chacune des relations dont il dépend) et la situation de chacune des règles figurant dans sa base des règles, selon une distance euclidienne pondérée par ses enjeux. Si cette distance est inférieure à un *seuil* dont la valeur est déterminée en fonction de son discernement (cf. équation 4.2), cette règle est considérée comme applicable dans la situation courante. Parmi ces règles applicables, l'acteur choisira aléatoirement l'une des trois règles dont la force est la plus élevée, et si aucune règle n'est applicable, il créera une nouvelle règle à partir de sa situation courante (cf. 4.1.2).

$$seuil = \text{distance}(\text{situationMax}, \text{situationMin}) / discernement \qquad \text{(éq. 4.2)}$$

où situationMax (respectivement situationMin) est la situation qui maximise (respectivement minimise) la satisfaction de l'acteur, et distance ( ) est une fonction qui retourne la distance euclidienne entre deux situations.

Le discernement d'un acteur, dont l'échelle de valeur est un entier compris entre 1 et 5, détermine comment l'acteur interprète cette notion de proximité et donc sa capacité à discriminer les situations. Une grande valeur (5) correspond à une perspicacité ou une sensibilité permettant à l'acteur d'évaluer et de distinguer assez précisément les situations telles que : très mauvaise, mauvaise, neutre, bonne, et très bonne. Tandis qu'avec une faible valeur (1), toutes les situations sont perçues comme similaires si bien que toutes les règles seront applicables dans toutes les situations et donc à chaque étape de la simulation.

La répartition du renforcement

L'apprentissage par renforcement attribue une récompense aux règles qui conduisent l'agent vers une « bonne » situation. Dans le jeu social, lorsqu'un acteur applique une règle, il en perçoit l'impact direct sur la variation de sa satisfaction à l'étape suivante de la simulation pour les relations qu'il contrôle, et il perçoit la façon dont les autres acteurs y réagissent deux étapes plus tard (cette perception étant en tout état de cause bruitée par l'effet des relations dont il dépend et qui sont contrôlées par des acteurs qui ne dépendent pas de lui). La mise à jour de la force d'une règle étant fonction de cette perception, il s'agit de déterminer dans quelle mesure la règle appliquée à l'étape t sera récompensée aux étapes t+1 ou t+2, autrement dit dans quelle mesure l'acteur prend en compte l'impact indirect de ses actions.

C'est dans ce but qu'est introduit le paramètre répartition. Il indique le pourcentage de rétribution de la règle appliquée à l'étape t qui lui sera attribué aux étapes t+1 et t+2. Ce paramètre permet de définir des acteurs qui ne prennent en compte que l'impact direct (i.e. rétribution majoritairement à t+1, donc en fonction de l'impact sur eux-mêmes) de leur action, ou au contraire, des acteurs qui raisonnent en fonction de l'effet de leurs actions sur les autres acteurs dont ils dépendent (i.e. rétribution majoritairement à t+2). La valeur par défaut simple à déterminer consiste à appliquer à l'étape t+1 la part des enjeux que l'acteur place sur les relations qu'il contrôle et le

restant à l'étape suivante. Cependant, cette façon de faire survalorise l'effet indirect qui incorpore le « bruit » des impacts provenant des acteurs qui ne dépendent pas de l'acteur considéré.

4.2.2 – Les principales variables de l'algorithme

Dans cette section, nous allons introduire les principales variables de l'algorithme. Ces variables sont mises à jour selon les équations présentées ci-dessous.

La satisfaction d'un acteur

Cette variable évalue la possibilité de l'acteur d'accéder à l'ensemble des relations dont il a besoin pour atteindre ses objectifs, pondérée par son besoin de ces relations. En d'autres termes, elle mesure, pour un acteur a, sa capacité à disposer des moyens nécessaires à la réalisation de ses objectifs.

$$satisfaction_t = satisfaction(e) = \sum_{r \in R} enjeu(r) \times effet_r(e_r) \qquad \text{(éq. 4.3)}$$

L'écart entre la satisfaction et l'ambition d'un acteur

Cette variable correspond à l'écart en proportion entre la satisfaction et l'ambition de l'acteur. Il s'agit d'une proportion mesurant la part de satisfaction dont l'acteur dispose par rapport à son ambition, calculée de la façon suivante :

$$écart_t = \frac{ambition_{t-1} - satisfaction_t}{ambition_{t-1} - satisfactionMin} \qquad \text{(éq. 4.4)}$$

où satisfactionMin est la somme des valeurs minimales des impacts de relations sur l'acteur, calculée de la façon suivante :

$$MinSatisfaction = \sum_{r \in R} \min_{e_r}(impact(r, e_r)) \qquad \text{(éq. 4.5)}$$

La valeur de l'écart est toujours inférieure ou égale à 1 et devient négatif lorsque l'acteur est satisfait (sa satisfaction dépasse son ambition. Dans le cas particulier où $ambition_{t-1}$ est égale à satisfactionMin, l'écart est égal à 0.

Cette grandeur va diminuer avec l'augmentation de la satisfaction de l'acteur et avec la diminution de son ambition. Elle évalue donc en pourcentage la distance entre la satisfaction et l'ambition actuelles de l'acteur.

L'ambition d'un acteur

Rappelons que l'ambition est le niveau de satisfaction visé par l'acteur et qu'elle varie au cours du temps. Elle est initialisée à la satisfaction maximale de l'acteur, qui est la somme des valeurs maximales des impacts de relations sur l'acteur.

$$MaxSatisfaction = \sum_{r \in R} \max_{e_r}(impact(r, e_r)) \qquad \text{(éq. 4.6)}$$

L'ambition d'un acteur varie en fonction de la réactivité, du taux d'exploration (TX, voir ci-après) et de l'écart de l'acteur. Elle est mise à jour de la façon suivante :

- Si l'acteur n'est pas satisfait, son ambition diminue proportionnellement à son écart et son taux d'exploration. Dans la phase d'exploration (un taux d'exploration élevé), on considère que l'acteur cherche de nouvelles actions lui permettant d'améliorer sa situation : son ambition diminue légèrement. Par contre, dans la phase d'exploitation (un taux d'exploration faible), on considère que l'acteur exploite les règles qu'il a déjà apprises, est

presque satisfait de sa situation, et a intérêt à stabiliser son comportement : son ambition se rapproche plus rapidement de sa satisfaction.

$$ambition_t = ambition_{t-1} - ((1 - TX_{t-1}) \times (réactivité\ /100) \times écart_t) \quad \text{(éq. 4.7)}$$

- Si l'acteur est satisfait, plus l'écart entre sa satisfaction et son ambition augmente, plus son ambition augmente, mais cependant pas au point de dépasser sa satisfaction si cette dernière ne diminue pas : si un acteur est satisfait, c'est parce qu'il a réalisé ses objectifs, il n'y a aucune raison qu'il devienne insatisfait si satisfaction demeure constante.

$$ambition_t = ambition_{t-1} + ((satisfaction_t - ambition_{t-1}) \times (réactivité\ /100)) \quad \text{(éq. 4.8)}$$

Plus un acteur est réactif, plus la variation de son ambition est importante.

Le taux d'exploration d'un acteur

Le taux d'exploration d'un acteur détermine la façon dont il va :

- réviser son ambition,
- mettre à jour la force des règles précédemment appliquées,
- choisir l'intensité de l'action d'une nouvelle règle qu'il crée, lorsqu'aucune n'est applicable.

Sa valeur est comprise entre 0,1 (forte exploitation) et 0,9 (forte exploration). Les valeurs 0 et 1 sont exclues : un taux nul correspondrait à un acteur qui n'explore pas du tout, et donc n'apprend rien ; un taux égal à 1 correspondrait à un acteur qui ne considère que les informations les plus récentes, et donc n'a pas de mémoire.

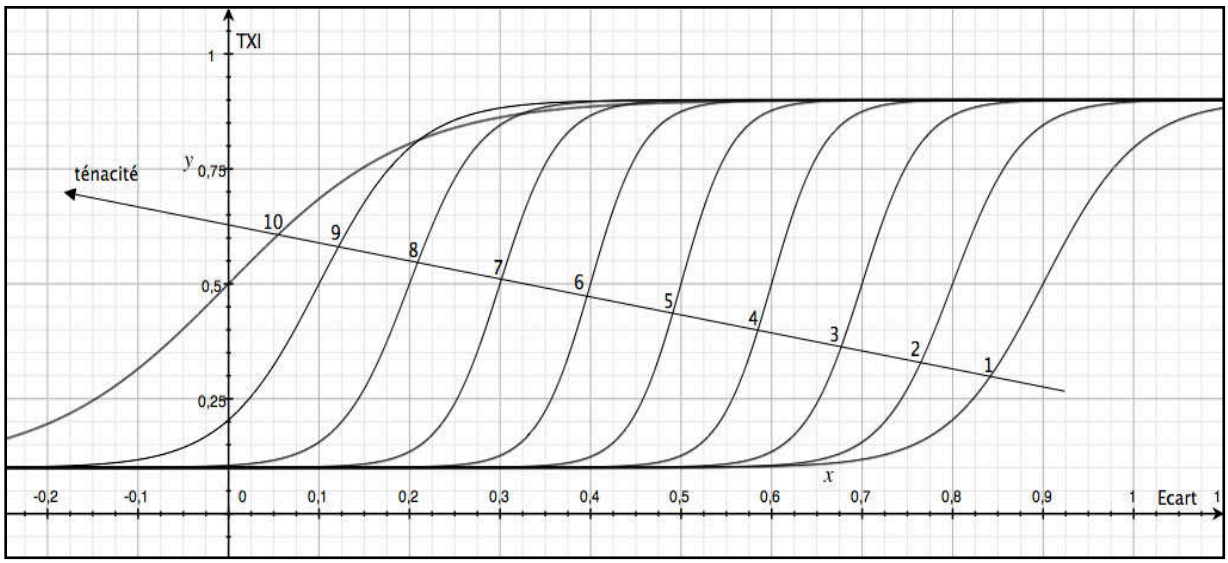


Figure 4.1. *La valeur du Taux d'Exploration Instantané (TXI_t) en fonction de l'écart et selon la valeur (de 1 à 10) de la ténacité de l'acteur.*

A chaque étape de la simulation, chaque acteur calcule un *taux d'exploration instantané* en fonction de son écart et de sa ténacité (cf. équations 4.9, 4.10, et 4.11). Ce taux d'exploration instantané varie selon une sigmoïde (cf. figure 4.1) : plus l'écart de l'acteur est faible, plus son taux d'exploration instantané est faible (un acteur satisfait de sa situation cherchera à la consolider). A l'inverse, plus l'écart est important, plus l'acteur aura tendance à explorer afin de trouver comment réduire cet écart. Les paramètres de cette sigmoïde dépendent de la ténacité de l'acteur : plus un acteur est tenace, plus il a tendance à explorer.

$$TXI_t = 0.1 + (\frac{0.8}{1 + e^{pente \times (écart_t - abscisse)}}) \quad \text{(éq. 4.9)}$$

$$\text{Où } pente = -(ténacité \times (10 - ténacité) + 10) \quad \text{(éq. 4.10)}$$

$$\text{Et } abscisse = (10 - ténacité) / 10 \quad \text{(éq. 4.11)}$$

La pente de la sigmoïde est donc symétrique pour les valeurs de ténacité de part et d'autre de 5 (*i.e.* ((5+x) × (10-(5+x)) = ((5-x) × (10-(5-x))). La variable écart varie entre –ε et 1. La sigmoïde n'est symétrique par rapport à l'axe y de l'écart que pour une ténacité égale à 10, et elle se décale vers les valeurs positive proportionnellement à la diminution de la ténacité, selon la valeur du terme (10 – ténacité)/10.

Le *taux d'exploration* d'un acteur est initialisé à la valeur du taux d'exploration instantané à l'étape 0, et il sera mis à jour en fonction du taux d'exploration instantanée et de la réactivité de l'acteur : plus un acteur est réactif, plus il attachera d'importance à son taux d'exploration instantané, et plus la variation de son taux d'exploration suivra de près la variation de son écart de la façon suivante :

$$TX_t = (1 - réactivité/10) \times TX_{t-1} + réactivité/10 \times TXI_t \quad \text{(éq. 4.12)}$$

L'intensité des actions

A l'étape 6 de l'algorithme 4.1, lorsqu'un acteur n'a aucune règle applicable, il doit créer une nouvelle règle. Les actions de cette règle sont choisies au hasard dans un intervalle dont les bornes sont définies par une variable, l'intensité des actions.

Avoir un comportement exploratoire consiste à appliquer une action énergique pour modifier sensiblement l'état du jeu. C'est pourquoi l'intensité des actions est d'autant plus grande que le taux d'exploration est important. Elle ne peut être nulle afin de préserver la capacité d'action de l'acteur. La valeur de l'intensité varie, dans l'intervalle [0,2 ; 1,8], linéairement par rapport au taux d'exploration, de la façon suivante :

$$intensité_t = 2 \times TX_t \quad \text{(éq. 4.13)}$$

Le coefficient 2 détermine la différence d'échelle de valeur entre le taux d'exploration et les actions. Plusieurs coefficients ont été testés dans l'intervalle [0 ; 20][13]. Une intensité élevée (coefficient proche de 20) n'est pas plausible car il est difficile de concevoir un acteur coopératif qui devient non-coopératif à l'étape suivante ; le changement doit s'effectuer progressivement. De même, une intensité faible (coefficient proche de 0) donne lieu à des actions très faibles, voire négligeables. Le coefficient 2 donne les résultats les plus plausibles.

Pour chacune des relations contrôlées par l'acteur, la variation de l'état de cette relation sera calculée de la manière suivante :

$$action_t = intensité_t \times random\ (\ ) \in [-1,8 ; -0,2] \cup [0,2 ; 1,8] \quad \text{(éq. 4.14)}$$

où random ( ) retourne une valeur aléatoire réelle entre -1 et 1.

La force des règles

La force d'une règle évalue son efficacité dans la poursuite de l'objectif de l'acteur : parmi les règles applicables dans une situation donnée, l'acteur choisira l'une de celles dont la force est la plus grande. Plus précisément, la force d'une règle évalue l'effet des applications passées de cette règle sur la variation de la satisfaction de l'acteur : c'est une agrégation de ce que la satisfaction de l'acteur a gagné ou perdu du fait de l'application de la règle. Initialisée à 0 au moment de la création

[13] L'amplitude de l'intervalle d'état d'une relation est égale à 20.

de la règle, elle est mise à jour après chaque application de la règle en fonction du taux d'exploration et de la variation de la satisfaction de la façon suivante :

$$RA_{t\text{-}1}.force_t = (1 - TX_t) \times RA_{t\text{-}1}.force_{t\text{-}1} + TX_t \times PRR \times \Delta satisfaction_t \quad \text{(éq. 4.15)}$$

$$RA_{t\text{-}2}.force_t = RA_{t\text{-}2}.force_{t\text{-}1} + TX_t \times (1 - PRR) \times \Delta satisfaction_t \quad \text{(éq. 4.16)}$$

où RA_h est la règle appliquée à l'instant h, PRR est le pourcentage de renforcement attribué à la dernière règle, et $\Delta satisfaction_t$ est la variation de la satisfaction de l'acteur :

$$\Delta satisfaction_t = satisfaction_t - satisfaction_{t\text{-}1} \quad \text{(éq. 4.17)}$$

En conséquence, plus l'acteur va explorer, plus son taux d'exploration est important, plus il va prendre en considération l'effet des actions des règles ($\Delta satisfaction$), et plus la force de la règle appliquée sera modifiée. Inversement, plus l'acteur exploite, plus son taux d'exploration est faible, moins la force de la règle appliquée sera modifiée.

4.2.3 – Relations entre les paramètres et les variables de l'algorithme

La figure 4.2 présente une schématisation de l'ensemble des interdépendances entre les paramètres (à droite) et les variables de l'algorithme. Une flèche d'un paramètre ou une variable vers une variable indique que la cible dépend de l'origine, soit positivement (flèche étiquetée par un +) soit négativement (-). Par exemple, l'*écart* de l'acteur varie dans le même sens que son *ambition*, et dans le sens inverse de sa *satisfaction*. Les flèches étiquetées par +' (respectivement par -') indique une influence positive (respectivement négative) de l'origine sur le taux de variation de la cible. Par exemple, plus le *taux d'exploration* de l'acteur augmente, plus la variation de son *ambition* diminue.

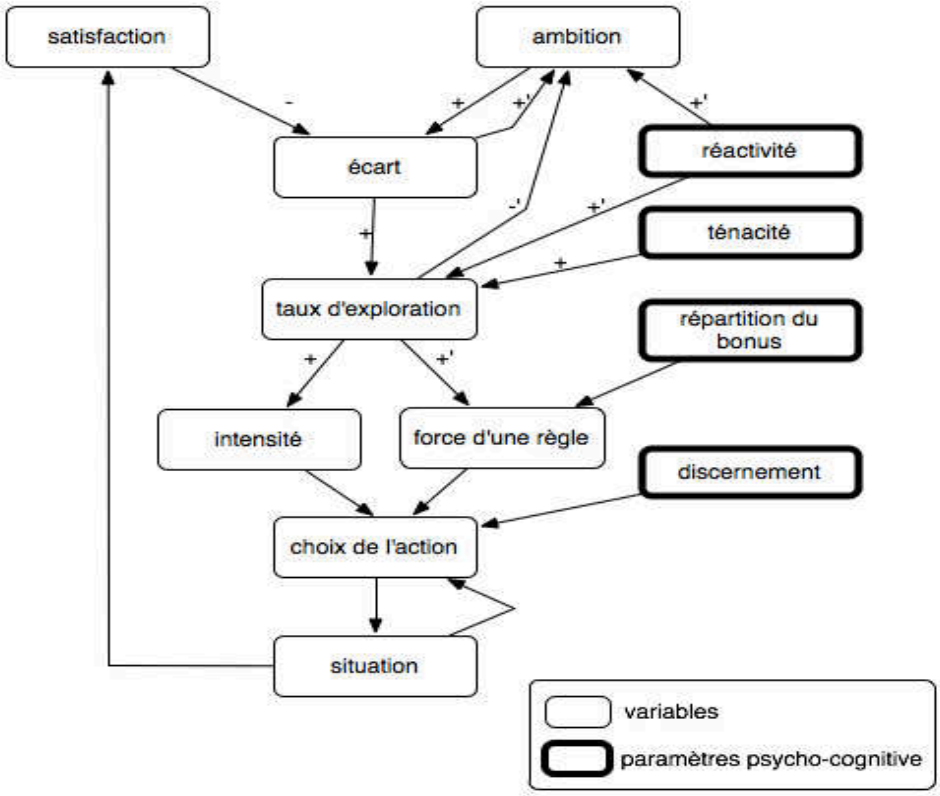


Figure 4.2. *Relations de dépendance entre les variables et les paramètres de l'algorithme. Les arcs étiquetés "+" (resp. "-") indiquent une variation de même sens (resp. sens contraire) ; ceux étiquetés "+'" ou "-'" indiquent une influence sur le taux de variation, la dérivée, de la cible.*

Le tableau 4.1 résume le sens de variation de certaines variables (en lignes) en fonction de la valeur de certains paramètres ou de l'augmentation de certaines variables (en colonne). Par exemple, plus la réactivité (en colonne) d'un acteur est importante, plus la variation de son taux d'exploration (en ligne) augmente[14].

	ténacité ⊕	réactivité ⊕	taux d'exploration ↗	satisfaction ↗	écart ↗
Taux d'exploration	↗			↘	↗
Δ taux d'exploration		↗			
force				↗	
Δforce			↗		
intensité			↗		
Δambition		↗	↘		↗

Tableau 4.1. *Sens de variation de certaines variables de l'algorithme (en ligne) en fonction de la valeur de certaines paramètres ou variables (en colonne). ⊕ correspond à une valeur du paramètre très importante, et ↗ indique le sens de variation de la variable.*

4.2.4 – L'algorithme de délibération d'un acteur

Initialisation

L'état de chaque relation est initialisé de façon arbitraire, par exemple à la valeur 0 qui correspond au comportement « neutre » de l'acteur qui contrôle la relation. La valeur des paramètres psycho-cognitifs (la ténacité, la réactivité, le discernement et la répartition du renforcement) de chaque acteur est définie par le modélisateur.

Les étapes de l'algorithme

Les étapes de l'algorithme sont les suivantes :

[14] Les cases vides indiquent qu'il n'y a pas de relation. Par exemple, l'augmentation de l'écart d'un acteur n'a aucun effet direct sur l'intensité des actions.

Initiation :

La satisfaction est calculée en fonction des états des relations dont l'acteur dépend (éq. 4.3)

L'ambition est initialisée à la valeur maximale de la satisfaction de l'acteur (éq. 4.6)

L'écart est calculé en fonction de la satisfaction et l'ambition (éq. 4.4 et 4.5)

Le taux d'exploration est initialisé à la valeur du taux d'exploration instantané (éq. 4.9, 4.10, et 4.11)

A chaque étape t de la simulation, l'acteur :

1. perçoit sa satisfaction (éq. 4.3), calcule son écart (éq. 4.4), et met à jour son ambition (éq. 4.7 et 4.8).

2. met à jour son taux d'exploration (éq. 4.9, 4.10, 4.11, et 4.12).

3. met à jour l'intensité des actions (équation 4.13).

4. met à jour la force des deux dernières règles appliquées (éq. 4.15, 4.16, et 4.17). Les règles de force négative sont oubliées.

5. sélectionne les règles applicables, celles dont la composante situation est proche de sa situation courante en fonction de son discernement (4.2).

6. Si l'ensemble des règles applicables est vide (notamment au début de la simulation), il crée une nouvelle règle, avec une force initialisée à 0, une situation égale à la situation courante et les actions (modifications à apporter sur les états des relations contrôlées) choisies au hasard dans l'intervalle [- intensité ; + intensité] (éq. 4.14).

7. choisit la règle nouvellement créée ou, parmi celles applicables, l'une de trois règles dont la force est la plus grande.

Lorsque tous les acteurs ont choisi une règle, leurs actions sont appliquées.

Algorithme 4.2. *Algorithme de délibération d'un acteur.*

4.2.5 – Discussions

Dans ce paragraphe, nous discutons l'implémentation de quelques étapes de l'algorithme. Le choix d'appliquer de telle ou telle méthode est parfois difficile, nous donnerons les arguments en faveur de la solution retenue.

L'action nulle

Une action nulle correspond à une règle dans laquelle l'acteur ne fait pas varier l'état des relations qu'il contrôle, c'est-à-dire $action_t = (0)$ dans l'équation 4.14. Cette action est interdite pour deux raisons. Tout d'abord, un acteur qui fait une action nulle est un acteur qui devrait être satisfait de sa situation et donc n'a pas intérêt à la changer, ce qui n'est pas nécessairement le cas.

De plus, l'équivalent de l'action nulle a été pris en considération dans le paramètre « fréquence » d'une relation. En effet, ce paramètre considère le cas où un acteur ne peut pas accéder à sa relation à chaque instant (pararaphe 2.2.1 du chapitre 2), et donc ne peut pas changer son état.

La gestion des limites de l'état d'une relation

A chaque étape de la simulation, un acteur modifie l'état des relations qu'il contrôle d'une certaine valeur, son action. Dans certains cas, l'application de cette action peut conduire à un état qui dépasse les limites : les bornes inférieures ou supérieures d'une relation ; c'est-à-dire $e_r + a_r < e_{min}$ ou bien $e_{max} < e_r + a_r$, où e_r est l'état de la relation r, a_r est l'action à appliquer à l'état de la

relation r, et e_{min} (respectivement e_{max}) est la borne inférieure (respectivement supérieure) de la relation. Pour palier ce problème, plusieurs solutions ont été testées. La première solution consiste à interdire les actions qui dépassent les limites et à forcer l'acteur à choisir une nouvelle action. Cela force l'acteur, dans la plupart des cas, à changer son comportement (en pratique choisir une action de signe opposé), ce qui n'est pas souhaitable : si l'état de la relation est proche d'une borne, c'est presque certainement parce que cet état est satisfaisant pour l'acteur, et il n'a donc pas de raison de s'en éloigner. Une deuxième idée est de remplacer l'action de la règle par une nouvelle action résultant de la troncature de l'ancienne action. Dans la plupart des cas, la troncature de l'action conduit à une action négligeable, et même parfois nulle, ce qui devient incohérents si les limites de la relation changent[15]. Une dernière solution, qui est celle qui a finalement été retenue, consiste à appliquer la troncature de l'action, mais sans modifier cette action dans la base des règles. La règle est alors récompensée comme si elle était appliquée alors qu'elle ne l'est que partiellement, voire pas du tout dès que l'état de la relation est égale à l'une des deux bornes. L'état de cette relation restera alors stationnaire tant que cette règle sera appliquée.

Au delà de l'ambition

Un acteur n'a pas de raison particulière de changer de comportement si sa situation est bonne, par contre il tentera de faire autre chose s'il estime que sa situation pourrait s'améliorer. On peut aussi considérer, ce que nous faisons, qu'un acteur continue à agir même s'il est content de sa situation, il cherchera à tenter de l'améliorer encore. En d'autres termes, dans notre contexte, un acteur n'est pas en mesure de détecter s'il peut améliorer sa situation quand il est satisfait. C'est pourquoi nous avons permis à un acteur de continuer à agir au delà de son ambition, ce qui lui permet de continuer à chercher à améliorer sa situation tant que la simulation continue.

L'oubli

L'oubli est le processus consistant à diminuer régulièrement la force de toutes les règles avec le temps. Le processus peut être appliqué à chaque étape ou après un certain nombre de pas fixé par le modélisateur. L'hypothèse sous-jacente est que les chances de maintenir la pertinence d'une règle diminuent avec le temps. Ce processus était essentiel dans la version de l'algorithme de simulation, proposée par [Mailliard, 2008], dont nous n'approfondirons pas l'étude dans ce mémoire. Dans cet algorithme, à chaque pas de la simulation, l'oubli était forfaitaire, ainsi que le renforcement de la force d'une règle, en fonction de deux paramètres « bonus » et « oubli » fixés par le modélisateur au début de la simulation.

Dans l'algorithme présenté ici, nous supposons qu'un acteur n'a aucune raison d'oublier ou d'affaiblir la force d'une « bonne » règle systématiquement avec le temps, sa force sera affaiblie et elle sera oubliée si et seulement si elle n'est plus pertinente. Le renforcement de la force de cette règle est effectué, d'une part en fonction de l'application de la règle, plus précisément en fonction de la variation de la satisfaction de l'acteur à la suite de l'application de la règle, et d'autre part en fonction du taux d'exploration. Cela permet à l'acteur de renouveler rapidement ses règles s'il explore, et de ne garder que les meilleures (celles qui apportent une grande variation positive de sa satisfaction). Et s'il exploite, il ne garde que les règles « pertinentes » lui permettant de se stabiliser dans une bonne situation : si une règle n'est plus pertinente sous certaines conditions, elle sera rapidement oubliée.

4.2.6 – Caractérisation des résultats de l'algorithme

Cet algorithme, tout en respectant l'hypothèse de la rationalité limitée, produit des résultats proches des optima de Pareto, et présente la propriété de converger rapidement lorsque la structure

[15] Les limites d'une relation changent dans le cas où l'état de cette relation est contraint par une autre relation.

du jeu valorise la coopération, et de ne pas converger ou de converger mal (une simulation longue et des satisfactions faibles) dans le cas contraire. Cette propriété est particulièrement pertinente, car elle permet de mettre un modèle d'organisation à l'épreuve de la régulation, et de déterminer si une organisation conçue *a priori* pour favoriser la coopération, la valorise effectivement. Plus particulièrement, dans le cas des jeux à somme nulles, si ce que l'un des acteurs gagne est perdu par un autre, il est impossible qu'un état procure une « bonne » satisfaction à tous les acteurs, ce qui mène les acteurs à chercher un compromis, au lieu de maximiser leur satisfaction, pour que le jeu puisse se stabiliser, et donc perdurer. Expérimentalement, on constate que l'algorithme nécessite un grand nombre de pas avant de converger dans le cas de tels jeux (cf. la variante 5/5 du dilemme de prisonnier classique, que nous étudions au paragraphe 4.4.1 ci-dessous).

D'une façon plus générale, la prolongation d'une simulation est un symptôme de la difficulté des acteurs à trouver comment coopérer. Cette difficulté peut provenir de la structure du jeu donne aux acteurs un faible avantage à coopérer (et même un avantage négatif dans le cas des jeux à somme nulle), ce que le sociologue pourra alors interpréter comme une fragilité de cette organisation. Cette difficulté peut aussi être propre à une simulation. Dans ce cas, les simulation les plus longues sont généralement celles qui procurent une moins bonne satisfaction aux acteurs, comme illustré dans la variante 3/7 du dilemme de prisonnier classique et surtout celle de 4/6 (cf. tableau 4.3 ci-dessous). Tout se passe comme si lorsque les acteurs ne trouvent pas assez rapidement comment coopérer, le jeu finit par se stabiliser mais sur un compromis peu satisfaisant pour l'ensemble.

D'autre part, il ne faut pas s'attendre à ce que le jeu converge systématiquement vers un optimum global, c'est-à-dire un état du jeu qui maximiserait la somme ou le produit des satisfactions des acteurs. Il n'est pas dit qu'une telle configuration soit socialement faisable, par exemple si elle pénalise excessivement l'un des acteurs. De plus, [Barel, 1979] montre que certaines organisations sont porteuses de plusieurs « potentialités », c'est-à-dire de plusieurs modalités selon lesquelles elles pourraient fonctionner, bien qu'une seule de ces modalités soit empiriquement observée (celle qui est « actualisée »). C'est effectivement ce que l'on observe dans certaines organisation, notamment celle que nous avons dénommée free-rider (cf. 2.3.3 et 4.4.3). Ce phénomène est bien mis en évidence par la répartition des points dans une analyse en composantes principales des résultats de simulation.

4.3 – Les méthodes d'analyse statistique utilisées

La dimension stochastique de cet algorithme de simulation conduit à l'exécuter plusieurs fois pour réaliser une expérience de simulation. Il est donc assez naturel d'utiliser des outils de l'analyse des données pour interpréter les résultats ; le format des résultats de simulation et d'analyse de sensibilité sera traité dans le chapitre 6 - SocLab.

L'étude statistique des résultats a un double objectif. D'une part, elle vise à découvrir des liens entre les variables. Et d'autre part, elle permet de réduire la taille de l'espace des données pour n'en conserver que les plus explicatives. Plus précisément, cette étude va mettre en évidence certaines propriétés d'un modèle d'une organisation. Par exemple, elle permet d'étudier comment le nombre de pas peut influencer les résultats obtenues en dispersion, en valeur, ou encore en corrélation.

On distingue les résultats relatifs aux relations (les valeurs d'états) des résultats relatifs aux acteurs (les satisfactions et les ambitions). Les valeurs de satisfactions des acteurs se déduisant analytiquement des valeurs des états des relations, il est inutile d'étudier les corrélations entre les variables des relations et les variables des acteurs. Nous nous concentrons donc sur les corrélations entre les variables des acteurs, les corrélations entre les variables des relations, l'allure de leurs distributions et l'influence de la longueur des simulations.

Tout d'abord, nous abordons le cas de l'analyse d'une seule variable. Ensuite nous discutons le cas de l'analyse bivariée de données. Et nous terminons par le cas de l'analyse multivariée.

4.3.1 – Analyse Univariée des données

L'analyse univariée étudie la distribution de chaque variable, indépendamment des autres. Elle permet de décrire et synthétiser les résultats en analysant une seule variable à la fois. Par exemple, la dispersion des valeurs de la satisfaction d'un acteur fait apparaître une position fragile de l'acteur au sein du modèle et donc de l'organisation. Une forte variance de l'état d'une relation offre de nombreuses interprétations : il peut s'agir d'une relation peu pertinente, dont les effets sur les acteurs sont peu importants.

Classiquement, l'analyse univariée utilise la moyenne, l'écart-type, éventuellement la médiane et les quartiles d'une variable représentés graphiquement par des histogrammes (cf. figure 4.3), qui sont un moyen rapide pour étudier la répartition d'une variable, et des boxplots (cf. figure 4.4) de la variable.

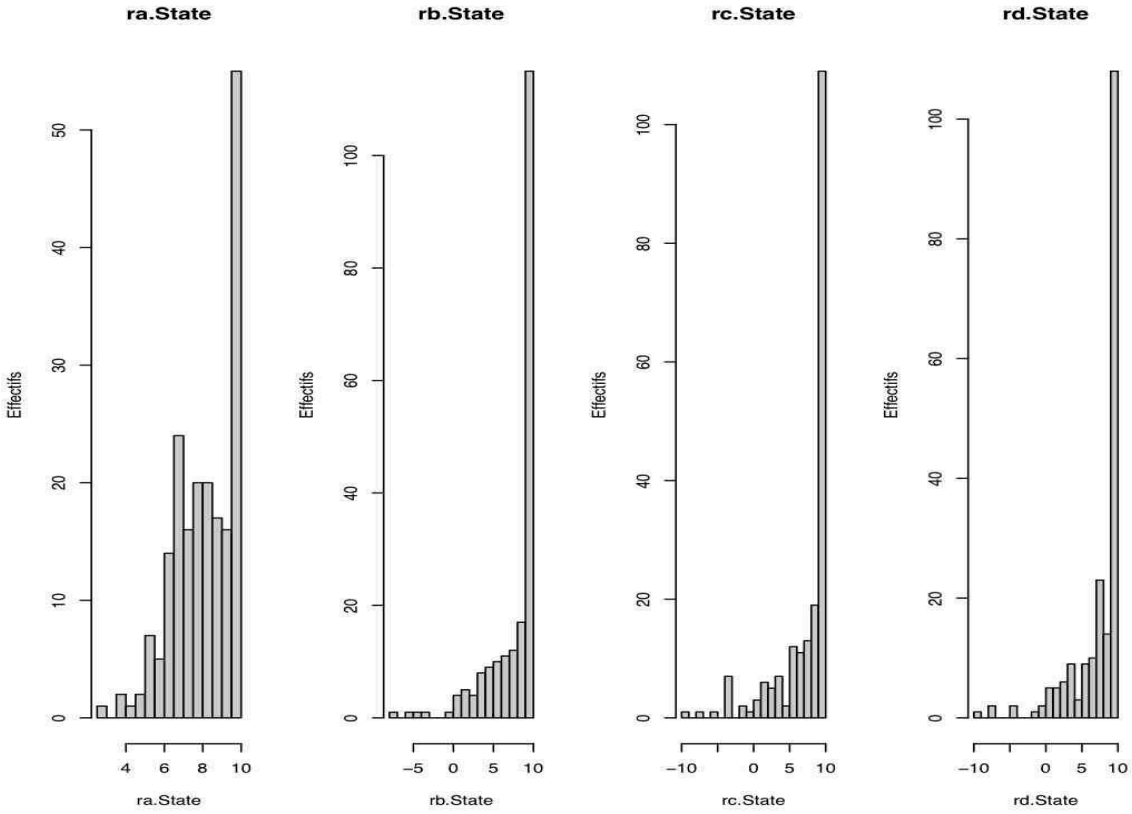


Figure 4.3. *Histogrammes des états des quatre relations (ra, rb, rc, et rd) du modèle free-rider résultant d'une expérience de simulation.*

Un boxplot (aussi appelé boite à moustaches) est un moyen de figurer le profil essentiel d'une série statistique quantitative. Il permet d'observer la dispersion et la répartition des valeurs que peut prendre une variable par rapport à ses quartiles. Un quartile est chacune des trois valeurs qui séparent la population des données en quatre parts égales, de sorte que chaque partie représente ¼ de la population. Le premier quartile (respectivement le second et le troisième) est la valeur de la variable en dessous de laquelle se trouve environ un quart (respectivement environ 50% et 75%) des individus de la population. Par exemple, dans la figure 4.4, plus que 75% des simulations se terminent avec un état de ra entre environ 6 et 10, et 25% des simulations se terminent avec un état de ra entre environ 9.5 et 10.

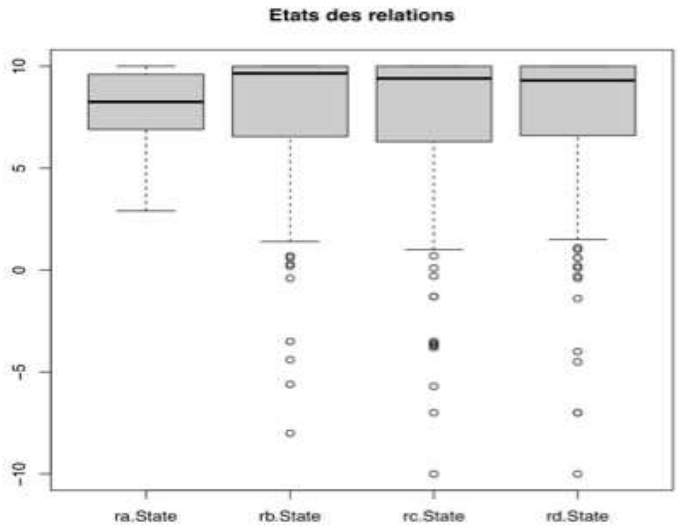


Figure 4.4. *Boxplot des états des quatre relations du modèle free-rider résultant d'une expérience de simulation.*

4.3.2 – Analyse Multivariée des données

L'analyse multivariée recouvre un ensemble de méthodes destinées à synthétiser l'information issue de plusieurs variables, pour mieux l'expliquer. Il s'agit de mettre en évidence les corrélations entre les variables, corrélations qui pourront éventuellement être interprétées comme des causalités par l'analyste. Ces éventuelles corrélations proviennent de la stratégie des acteurs dans l'ajustement de leur comportement, et ne peuvent pas être déduites des propriétés de la structure d'une organisation. Par exemple, une forte corrélation positive (respectivement négative) entre deux acteurs montre la possibilité d'une conjonction d'intérêts (respectivement d'un conflit structurel), alors que les positions structurelles de chacun n'ont aucun point commun et qu'ils n'entretiennent pas de solidarités. Cette corrélation met en évidence un « effet système » de l'organisation sur l'ajustement des comportements.

Il existe deux grandes catégories de méthodes : les méthodes descriptives, qui visent à structurer et simplifier les données issues de plusieurs variables, sans privilégier l'une d'entre elles en particulier. Et les méthodes explicatives, qui visent à expliquer une variable à l'aide de deux ou plusieurs variables explicatives.

Dans ce paragraphe, nous n'introduisons pas toutes les méthodes multi-variées des données, mais les trois méthodes que nous avons utilisées pour l'interprétation des résultats des simulations et des analyses de sensibilité [El Gemayel *et al.*, 2011]. [Popic, 2012][16] présente plusieurs programmes développés pour permettre l'analyse statistique des résultats de simulations.

L'Analyse en Composantes Principales

L'Analyse en Composantes Principales est une méthode descriptive, qui structure et résume l'information contenue dans les données, et procède de façon à ce que la perte d'informations résultante soit la plus faible possible. Elle consiste à extraire d'un ensemble de variables un nombre restreint de nouvelles variables indépendantes : les composantes principales. Ces composantes principales, qui ne correspondent pas à l'une ou l'autre des variables mais à une combinaison linéaire de variables, forment une base dans laquelle les données sont représentées de la façon la plus révélatrice possible de leur covariance.

Pour déterminer la pertinence d'une composante, on calcule la proportion d'inertie totale expliquée par (la valeur propre de) cette composante. L'inertie totale rend compte de la dispersion

[16] http://www.nathalievilla.org/spip.php?article86

des données de la population et est égale à la somme des variances des variables étudiées. Les composantes principales sont classées par ordre de pourcentage d'inertie totale expliquée décroissante. Généralement, on considère les premières composantes principales jusqu'à ce que le pourcentage cumulé d'inertie expliquée dépasse 80%.

Lorsque les deux premières composantes d'une ACP sont significatives, nous pouvons en donner deux représentations graphiques. La première est la projection des données dans un repère formé des deux axes associés aux deux premières composantes principales : elle permet de mettre en évidence des classes différentes dans la population des simulations, lorsqu'il en existe (4.8.a, 4.9.a, et 4.10.a). Une autre représentation graphique est la représentation des variables par rapport au cercle unité dans un repère dont les deux axes sont les deux premières composantes principales. Dans ce repère chaque variable initiale est représentée par un point dont les coordonnées mesurent sa contribution à chacun des deus axes (4.8.b, 4.9.b, et 4.10.b).

La Classification Ascendante Hiérarchique

La Classification Ascendante Hiérarchique (CAH) est une méthode descriptive qui procède de façon automatique au regroupement des individus similaires selon un critère de « dissimilarité » qui est calculé par la méthode MDS (Multi-Dimensional Scaling). MDS positionne un ensemble de N individus dans un plan à deux dimensions construit de telle manière que la distance euclidienne entre deux points dans le plan est approximativement égale à la valeur de leur dissimilarité (ou distance) dans l'espace initial dans lequel chaque variable constitue une dimension. Pour chaque couple, elle calcule un indice de dissimilarité qui sera enregistré dans une matrice dite « matrice de distance ». Plus l'indice est petit, plus la distance entre les individus est petite et donc plus les individus se ressemblent. Inversement, plus la distance est élevée, plus les individus sont différents.

La Classification Ascendante Hiérarchique procède par le regroupement deux à deux des individus les plus proches (cf. figure 4.5), selon la distance entre les individus. Puis, de façon itérative, elle agrège les éléments (individus ou groupes d'individus) les plus semblables. Le processus d'agrégation ne s'arrête que lorsque l'ensemble de tous les individus se retrouve dans un groupe unique. La classification est dite « ascendante » car elle part des observations individuelles (c'est-à-dire où tous les individus sont seuls dans un groupe). Le qualificatif « hiérarchique » vient du fait qu'elle produit des groupes de plus en plus vastes, incluant des sous-groupes en leur sein.

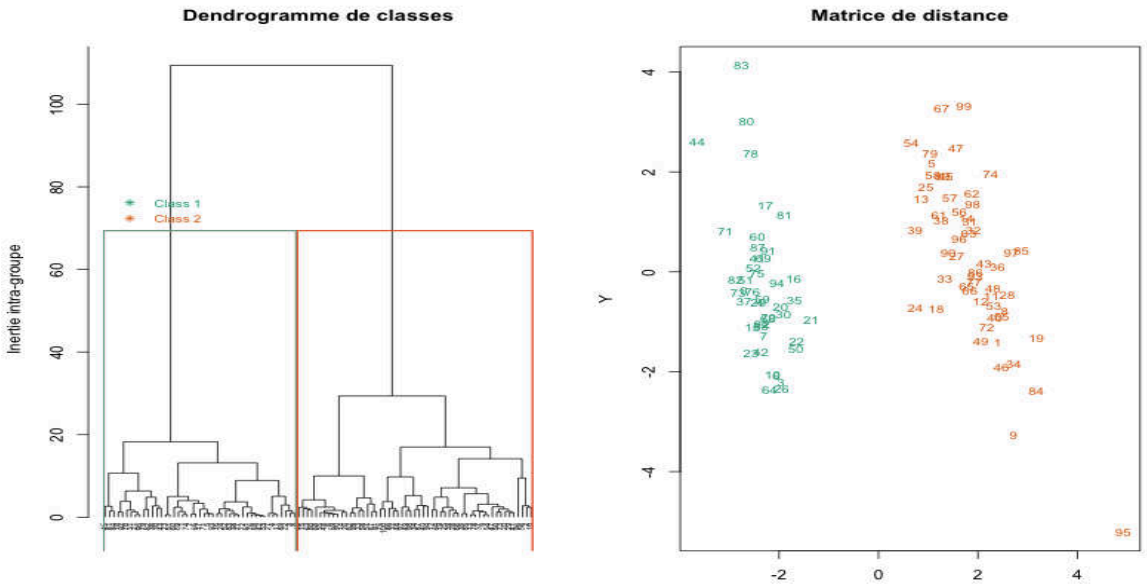


Figure 4.5. *Classification Ascendante Hiérarchique des résultats de simulations du cas Bolet (cf. 5.4.3), l'utilisateur a choisi une classification en deux groupes.*

Lors de son stage, Soraya Popic a créée une fonction permettant de réaliser une Classification Ascendante Hiérarchique de manière automatique sur les résultats de simulations produits par SocLab. L'utilisateur a la possibilité de spécifier interactivement le nombre de groupes qu'il souhaite considérer (la contrainte généralement fixée est de choisir un nombre de groupes minimal pour une inertie intra-groupe minimale).il peut alors obtenir un tableau d'analyse des moyennes (cf. par exemple le tableau 5.25) en format csv, ainsi que les boxplots des variables (Etat des relations et satisfaction des acteurs) dans les différents groupes spécifiés (cf. figures 5.20 et 5.21).

La régression linéaire

On distingue entre la régression linéaire simple, qui est appliquée sur deux variables, et la régression linéaire multiple, qui est appliquée sur plusieurs variables. Dans ce paragraphe, on se concentre sur le cas général : la régression linéaire multiple.

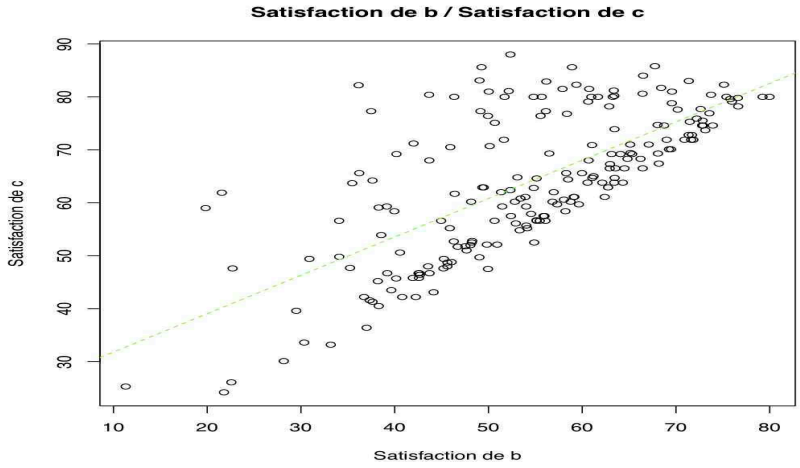


Figure 4.6. *Nuage de points et droite de régression linéaire entre les satisfactions des deux acteurs « b » et « c » du modèle free-rider.*

La régression linéaire (cf. figure 4.6) est une méthode explicative, qui permet d'expliquer une variable numérique y par une ou plusieurs autres variables numériques indépendantes. Elle cherche d'une part à trouver un bon modèle de prédiction des valeurs de y, et d'autre part à déterminer parmi les variables x_i celles dont l'effet est le plus significatif sur la valeur y. Formellement, elle permet de modéliser la relation entre une variable y et un vecteur de variables explicatives x. De manière générale, le modèle que produit une régression linéaire s'écrit de la manière suivante :

$$y = \sum_{i=0,\ldots,n} \alpha_i x_i + \beta$$

où y est la variable à expliquer, x_i les variables explicatives, β une constante, α_i les coefficients de régression partiels, et n est le nombre des variables explicatives.

Le nombre de variables explicatives n'est pas fixé dans cette méthode. On répète le processus de la régression linéaire après la suppression des variables explicatives les moins significatives jusqu'à ne garder que les variables significatives. La significativité d'une variable est estimée par sa *p-value* : plus celle-ci est faible plus la variable est significative.

L'appréciation de la qualité de la régression linéaire se fait grâce à deux indicateurs. Le premier est le coefficient de corrélation multiple R^2, qui mesure la liaison entre la variable à expliquer et les différentes variables explicatives : si sa valeur est inférieure à 0,85 la liaison est médiocre et le modèle de régression est peu satisfaisant. Le deuxième indicateur est le coefficient de détermination R^2 ajusté qui intègre le nombre de variables retenues dans le calcul du pourcentage de variation expliqué par les variables explicatives. Le R^2 ajusté permet de connaître le meilleur modèle n'utilisant qu'un nombre limité de variables.

4.4 – Analyse des résultats produits par l'algorithme

En l'absence d'observations neurophysiologiques sur les processus mis en œuvre par les acteurs sociaux pour déterminer leurs comportements les uns vis à vis des autres, la validation d'un tel algorithme est fondée sur les réponses aux questions suivantes :

- est-il compatible avec ce que disent les sciences humaines et sociales ?
- est-il bien structuré et documenté, de telle sorte que l'on puisse comprendre son fonctionnement ?
- empiriquement, les résultats qu'il fournit sont ils vraisemblables (Ahrweiler et Gilbert, 2005) ?

Comme élément de la validation de cet algorithme, nous allons analyser son comportement sur une série de modèles d'organisations virtuelles et réelles ; nous ne prétendrons pas que cette validation soit définitive mais que, sur de tels exemples, notre algorithme donne les résultats attendus. Nous étudierons essentiellement des organisations formelles, le dilemme du prisonnier et free rider, dont les résultats sont simples à interpréter.

4.4.1 – Le dilemme du prisonnier

La structure du jeu considéré dans ce paragraphe est celle présentée dans la section 2.3.1 au chapitre 2, dans laquelle on fera varier systématiquement la répartition des enjeux de chacun des acteurs de 5 à 0 sur la relation qu'ils contrôlent et le complément, de 5 à 10, sur la relation que l'autre acteur contrôle. Les fonctions d'effets des deux relations étant en miroir l'une de l'autre, ces jeux sont donc parfaitement symétriques.

Dans chacun de ces jeux, les satisfactions maximale et minimale que peut obtenir chacun des acteurs sont respectivement 100 et -100 et pour toute valeur V dans cet intervalle, il existe un état e du jeu tel que, pour l'acteur A, satisfaction(A, e) = V. Avec la répartition des enjeux 5/5 (5 d'enjeux sur chacune des deux relations), il s'agit d'un jeu à somme nulle : quelque soit l'état e du jeu, satisfaction (A, e) + satisfaction (B, e) = 0. Dans les autres cas, plus l'écart entre les deux enjeux des acteurs est important, plus l'amplitude entre les maximum et minimum de la satisfaction globale (i.e. la somme des satisfactions des deux acteurs) est importante : elle est de 80 (entre -20 et +20 pour chacun des deux acteurs) pour la répartition 4 / 6 et de 400 (entre -100 et +100 pour chacun des deux acteurs) pour la répartition 0 / 10. Cette amplitude de la satisfaction de chaque acteur détermine donc l'importance du bénéfice résultant de la coopération. Dans tous les cas, un comportement coopératif (ou non coopératif) de la part d'un acteur correspond à une valeur positive (ou négative) de l'état de la relation qu'il contrôle. Le maximum (minimum) social est donc obtenu pour la valeur 10 (-10) pour chacune des deux relations.

Chaque cas fait l'objet de 200 simulations qui convergent toutes. Les paramètres psycho-cognitifs sont initialisés de façon standard : discernement = 1, ténacité = réactivité = 5, et une répartition de SERC*10%[17] pour la dernière règle et ((10-SERC)*10) % pour l'avant dernière règle. Des extraits des résultats de simulation obtenus avec SocLab sont donnés dans le tableau 4.2 avec, pour chacune des répartitions d'enjeux considérées (1ère colonne), la satisfaction moyenne des acteurs à convergence (2ème colonne), l'état moyen des relations à convergence (3ème colonne). Les courbes montrent l'évolution de l'état de la relation ra à chaque pas (en abscisse) de chacune des simulations ; les signes ⊕ marquent l'état final de la relation à l'issue d'une simulation, et la courbe rouge isolée, en évidence sur la variante 4/6, correspond à la valeur moyenne de l'état sur toutes les simulations.

[17] SERC est la somme des enjeux posés sur les relations contrôlées par l'acteur. 10 – SERC est donc la somme des enjeux posés sur les relations contrôlées par les autres acteurs.

On peut en premier lieu remarquer que les résultats respectent la symétrie de la structure du jeu ; il n'y a donc pas de biais sur le nom ou l'ordre dans lequel sont introduits les acteurs et les relations. On obtient les mêmes résultats si l'on inverse le sens des fonctions d'effet (c'est-à-dire avec un comportement coopératif correspondant à la valeur -10 (au lieu de 10) de l'état des relations) ; le signe des valeurs n'introduit pas de biais.

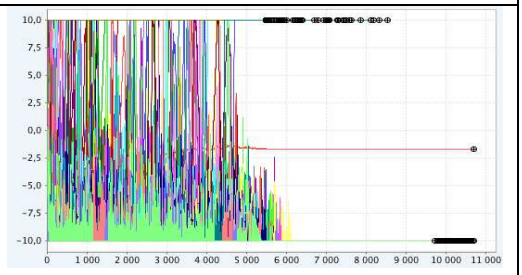
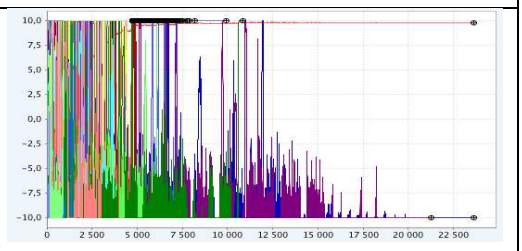
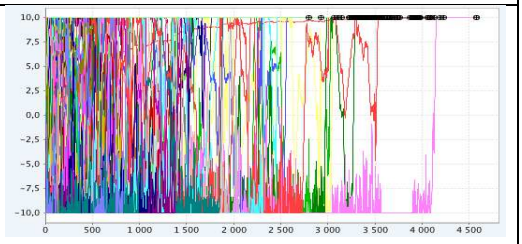
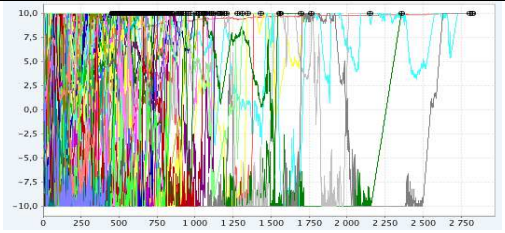
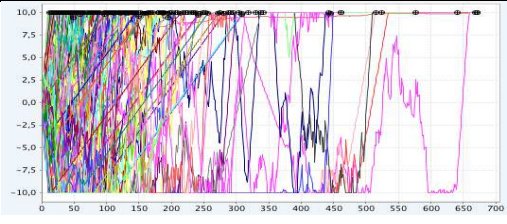
Enjeux	Satisfactions à convergence	Etat des relations à convergence	Trace de l'état de la relation rA pour 200 simulations
5 / 5	<table><tr><th></th><th>Moyenne</th><th>EcartType</th></tr><tr><td>A</td><td>0</td><td>0</td></tr><tr><td>B</td><td>0</td><td>0</td></tr></table>	<table><tr><th></th><th>Moyenne</th><th>EcartType</th></tr><tr><td>rA</td><td>-9,9</td><td>0,2</td></tr><tr><td>rB</td><td>-9,9</td><td>0,2</td></tr></table>	
4 / 6	<table><tr><th></th><th>Moyenne</th><th>EcartType</th></tr><tr><td>A</td><td>-3,4</td><td>19,42</td></tr><tr><td>B</td><td>-3,4</td><td>19,42</td></tr></table>	<table><tr><th></th><th>Moyenne</th><th>EcartType</th></tr><tr><td>rA</td><td>-1,7</td><td>9,71</td></tr><tr><td>rB</td><td>-1,7</td><td>9,71</td></tr></table>	
3 / 7	<table><tr><th></th><th>Moyenne</th><th>EcartType</th></tr><tr><td>A</td><td>39,2</td><td>1,58</td></tr><tr><td>B</td><td>39,2</td><td>1,58</td></tr></table>	<table><tr><th></th><th>Moyenne</th><th>EcartType</th></tr><tr><td>rA</td><td>9,8</td><td>0,4</td></tr><tr><td>rB</td><td>9,8</td><td>0,4</td></tr></table>	
2 / 8	<table><tr><th></th><th>Moyenne</th><th>EcartType</th></tr><tr><td>A</td><td>60</td><td>0</td></tr><tr><td>B</td><td>60</td><td>0</td></tr></table>	<table><tr><th></th><th>Moyenne</th><th>EcartType</th></tr><tr><td>rA</td><td>10</td><td>0</td></tr><tr><td>rB</td><td>10</td><td>0</td></tr></table>	
1 / 9	<table><tr><th></th><th>Moyenne</th><th>EcartType</th></tr><tr><td>A</td><td>80</td><td>0</td></tr><tr><td>B</td><td>80</td><td>0</td></tr></table>	<table><tr><th></th><th>Moyenne</th><th>EcartType</th></tr><tr><td>rA</td><td>10</td><td>0</td></tr><tr><td>rB</td><td>10</td><td>0</td></tr></table>	
0 / 10	<table><tr><th></th><th>Moyenne</th><th>EcartType</th></tr><tr><td>A</td><td>99,69</td><td>0.51</td></tr><tr><td>B</td><td>99,66</td><td>0,54</td></tr></table>	<table><tr><th></th><th>Moyenne</th><th>EcartType</th></tr><tr><td>rA</td><td>9,98</td><td>0,04</td></tr><tr><td>rB</td><td>9,96</td><td>0,07</td></tr></table>	

Tableau 4.2. *Extrait des résultats de simulation. La courbe rectiligne sur les graphes de l'évolution de l'état de la relation rA correspond à la moyenne.*

	0 / 10	1 / 9	2 / 8	3 / 7	4 / 6	5 / 5	6 / 4
Satisfaction min à max	-100 à 100	-80 à 80	-60 à 60	-40 à 40	-20 à 20	0 à 0	-20 à 20
Etat des relations	9,97	10	10	9,8	-1,7	-9,9	10
Ecart-type des états	0,055	0	0	0,4	9,71	0,2	0
Satisfaction des Acteurs	99,675	80	60	39,2	-3,4	0	20
% de satisfaction maximale conjointe	99,8%	100%	100%	99%	41,5%	100%	100%
Nombre de pas pour la convergence	131	700	3462	5749	8567	6219	5498

Tableau 4.3. *Tableau récapitulatif des résultats de 100 simulations en fonction de la répartition des enjeux des deux acteurs.*

Ces résultats sont synthétisés dans le tableau 4.3 dont les lignes s'interprètent de la façon suivante :

- Satisfaction min à max : dans chacun de ces jeux, pour chaque acteur la satisfaction maximale est de 100 (l'état étant à 10 pour la relation qu'il contrôle et à -10 pour l'autre relation) et celle minimale de -100 (répartition inverse de l'état des relations). Ce que l'on indique ici, c'est les valeurs min et max de la satisfaction que les deux acteurs peuvent avoir simultanément, c'est-à-dire leurs satisfactions respectives pour les valeurs min et max de la satisfaction globale. Plus l'écart entre les valeurs min et max est important, plus la coopération est bénéfique et son absence pénalisante.
- Etat des relations et satisfaction des acteurs : il s'agit de la valeur moyenne de l'état de chacune des deux relations et de la valeur moyenne de la satisfaction de chacun des acteurs sur les simulations.
- Ecart type de l'état des relations : cet écart type, la moyenne de celui des deux relations, est à rapporter à l'intervalle [-10, +10] du domaine de valeur de l'état des relations. Plus sa valeur est faible, plus les simulations donnent des résultats proches les unes des autres. Cela revient à dire que la régulation est plus forte, les acteurs adoptent le même comportement de façon plus systématique, comme s'ils n'avaient pas le choix.
- % de satisfaction : c'est le pourcentage de la satisfaction obtenue par rapport au maximum social ; le domaine de la satisfaction étant centré autour de 0, la formule est (satisfaction - satisfactionMin) / (satisfactionMax – satisfactionMin). Cette grandeur marque le taux de coopération de chacun des acteurs vis à vis de l'autre.
- Moyenne du nombre de pas nécessaires pour atteindre la convergence : cette grandeur correspond à la rapidité avec laquelle les acteurs obtiennent un niveau de satisfaction qui satisfait chacun d'eux.

En ce qui concerne le jeu avec une répartition 5 / 5 des enjeux, les simulations convergent vers l'équilibre de Nash ; dans un tel jeu, la coopération est coûteuse pour un acteur et elle est au bénéfice exclusif de l'autre acteur, tous les états du jeu étant équivalents du point de vue de la satisfaction globale. L'acteur perçoit rapidement (dans les premiers 5% du nombre de pas d'une simulation) que la coopération n'est pas bénéfique et ne lui permet pas d'augmenter sa satisfaction. Il cherche donc à stabiliser le jeu dans l'équilibre le moins couteux (le maximum des minimums de sa satisfaction), ce qui se traduit par des actions défensives préservant l'impact de la relation qu'il contrôle.

Le jeu avec la répartition 4 / 6 constitue un passage de la compétition entre les acteurs dans le jeu avec la répartition 5 / 5, dans lequel la coopération est au bénéfice exclusif de l'autre acteur, à la coopération quasiment complète dans le jeu avec une répartition 3 / 7, dans lequel la coopération est

bénéfique pour les deux acteurs. C'est un cas spécial : un acteur adopte la coopération quand qu'il la trouve profitable ; dans un tel jeu, les acteurs n'arrivent pas à la trouver à cause de la grande imprécision des informations sur les effets de leurs actions et à cause de leur faible ténacité. Il apparaît que la plupart des simulations convergent lentement vers une moindre coopération.

Malgré la coopération s'établit entre les deux acteurs dans le jeu avec une répartition 3 / 7, quelques simulations (2%) convergent lentement vers une moindre coopération.

Il apparaît clairement que, pour une répartition entre 3 / 7 et 0 / 10, plus les acteurs ont intérêt à coopérer, à leur détriment (sauf dans le cas 0 / 10) mais au bénéfice de la satisfaction globale du système, plus ils le font rapidement (nombre de pas pour la convergence).

Quant aux jeux dans lesquels la coopération (en terme de bonne valeur pour la satisfaction globale) est directement bénéfique pour celui qui la pratique, l'algorithme donne de très bons résultats, même si le nombre de pas pour la convergence est important.

Le graphique de la figure 4.7 montre que le nombre de pas pour atteindre une configuration stabilisée croit exponentiellement avec l'enjeu que l'acteur place sur la relation qu'il contrôle.

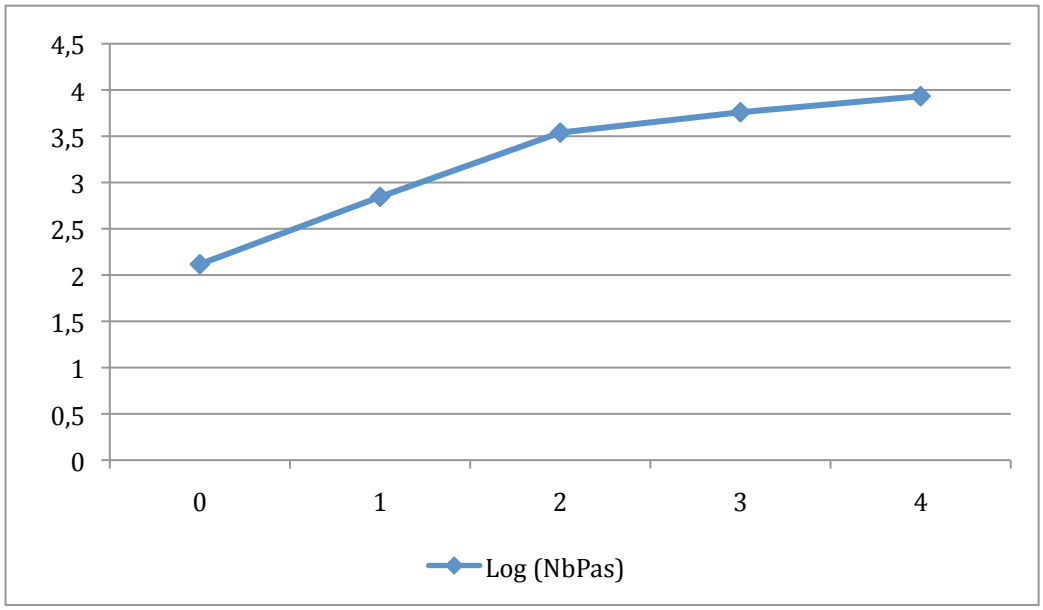


Figure 4.7. *Le logarithme du nombre de pas (axe y) en fonction de l'enjeu (axe x) que l'acteur place sur la relation qu'il contrôle.*

4.4.2 – Le dilemme du prisonnier à n acteurs

Le jeu considéré dans ce paragraphe est celui présenté au paragraphe 2.3.2 du chapitre 2, dans lequel on fera varier le nombre d'acteurs et de relations de 2 à 5. Ce jeu se présente comme un dilemme du prisonnier circulaire : Act1 dépend de Act2, Act2 dépend de Act3, …, Actn-1 dépend de Actn, et Actn dépend de Act1, où n est le nombre d'acteurs. La distribution des enjeux est de 1 sur la relation que l'acteur contrôle et de 9 sur l'autre relation.

Chaque cas fait l'objet de 100 simulations limitées à 1 000 000 pas. Les paramètres psycho-cognitifs sont initialisés de façon standard : discernement = 1, ténacité = réactivité = 5, et une répartition de 10% pour la dernière règle et 90% pour l'avant dernière règle.

	2	3	4	5
Satisfaction min à max	-80 à 80	-80 à 80	-80 à 80	-80 à 80
Etat des Relations	10	10	10	10
Ecart type de l'état des Relations	0	0	0	0
Satisfaction des Acteurs	80	80	80	80
% de Satisfaction des Acteurs	100%	100%	100%	100%
Nb pas pour la convergence	700	9572	49767	217862
% de convergence en moins de 1 000 000 pas	100%	100%	100%	100%

Tableau 4.4. *Résultats de 100 simulations pour le dilemme du prisonnier circulaire à n acteurs, où n = 2, …, 5.*

Les résultats sont synthétisés dans le tableau 4.4. Comme indiqué dans le paragraphe (4.1.5), la simulation se poursuit et s'arrête, en tout état de cause, lorsqu'est atteint le nombre maximal de pas que l'expérimentateur a défini. La dernière ligne indique que 100% des simulations ont atteint un état de convergence en moins de 1 000 000 pas.

Il apparaît clairement à l'examen de ces résultats que plus le nombre d'acteurs augmente, plus la coopération est difficile à trouver, et moins les acteurs coopèrent rapidement. Bien qu'en théorie le comportement des acteurs semble être très simple (chacun doit céder sa relation à un autre), il est difficile à trouver en pratique. En fait, tous les acteurs doivent coopérer en même temps (à un pas prêt) ce qui devient de plus en plus difficile avec l'augmentation du nombre d'acteurs.

Le graphique de la figure 4.8 montre que le nombre de pas pour atteindre une configuration stabilisée croit exponentiellement avec le nombre d'acteurs.

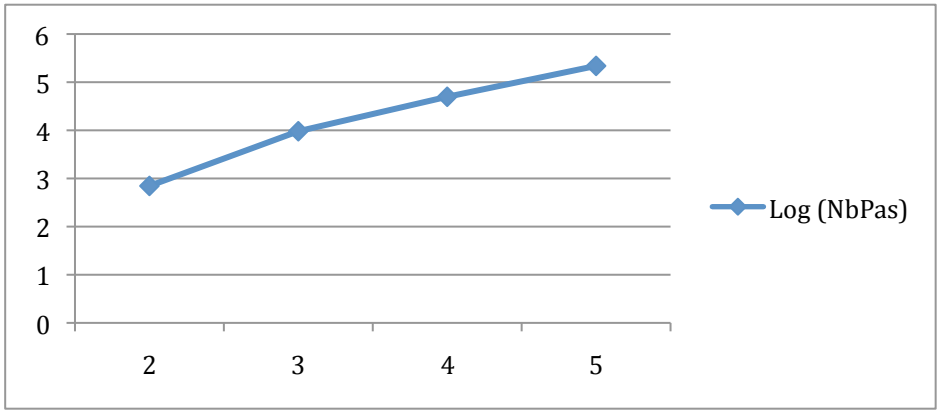


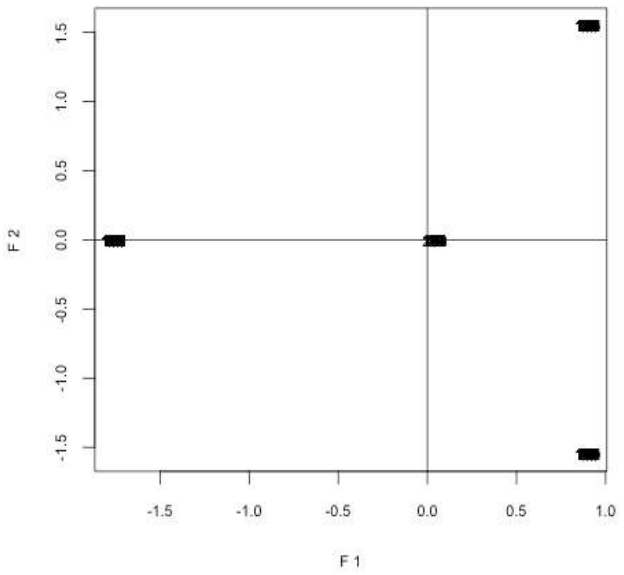
Figure 4.8. *Le logarithme du nombre de pas (axe y) en fonction du nombre d'acteurs (axe x).*

4.4.3 – Le modèle free-rider

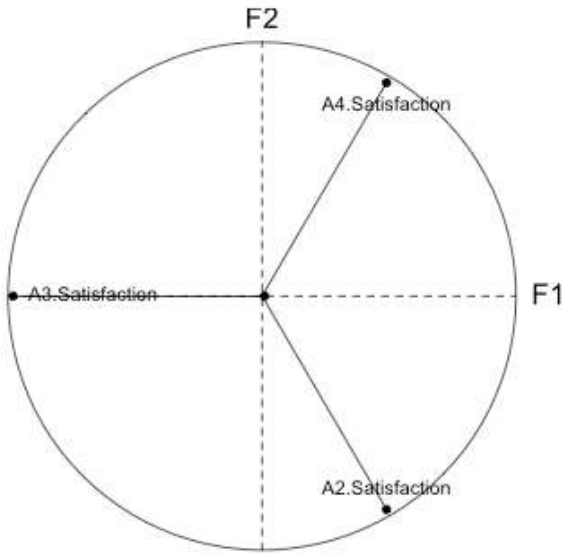
Le jeu considéré dans ce paragraphe est celui présenté au paragraphe 2.3.3 du chapitre 2. Les tableaux 4.5 et 4.6 et la figure 4.9 présentent les résultats de 200 simulations. Les paramètres psycho-cognitifs sont initialisés de façon standard : discernement = 1, ténacité = réactivité = 5, et une répartition de 10% pour la dernière règle et 90% pour l'avant dernière règle.

Configurations	C1	C2	C3	C4	C5	C6	C7	C8	C9
% d'occurrences	10,5	29,5	30,5	29,5	0	0	0	0	0

Tableau 4.5. *Le pourcentage d'apparition de chacun des neuf états remarquable (cf. tableau 2.3, chapitre 2) dans les résultats de simulation : l'état qui maximise la satisfaction globale (C1) ; les quatre états qui maximisent la satisfaction d'un des quatre acteurs : A1 (C8), A2 (C2), A3 (C3) et A4 (C4) ; trois états dans lesquels deux des trois acteurs ne coopèrent pas : A2 et A3 (C5), A2 et A4 (C6), A3 et A4 (C7) ; l'équilibre de Nash (C9).*



(a) représentation graphique des simulations dans un repère formé des deux axes associés aux deux premières composantes principales (F1 et F2) ; les (agglomérats) de points correspondant aux configurations C1 à C4 sont respectivement au centre, en bas, à gauche et en haut



(b) représentation graphique des variables (les satisfactions des quatre acteurs, celle de A1 est au centre) par rapport au cercle unité dans un repère dont les deux axes sont les deux premières composantes principales (F1 et F2)

Figure 4.9. *Analyse en composantes principales des résultats de simulation du modèle free-rider obtenue avec le logiciel R (71,26 de la variance est expliquée).*

Le tableau 4.6 caractérise chaque composante principale comme une combinaison linéaire des satisfactions des acteurs. Le premier Axe principal est une opposition entre la satisfaction de A3 et les satisfactions de A2 et A4. Tandis que le deuxième axe est une opposition entre la satisfaction de A2 et celle de A4. En global, il s'agit d'une opposition entre les satisfactions des trois acteurs ; un seul acteur parmi les trois peut parfois profiter de la coopération des deux autres et jouter la trahison.

	Satisfaction			
	A1	A2	A3	A4
F1	0,01	0,49	-0,98	0,49
F2	0	-0,84	0	0,84

Tableau 4.6. *Contribution des variables (les satisfactions des acteurs) aux deux composantes principales F1 et F2.*

Donnons une interprétation de ces résultats. Cette organisation peut être régulée et donc fonctionner selon quatre modalités différentes, qui correspondent aux quatre optima de Pareto donnant la satisfaction globale la plus élevée, et chacune est très précisément définie. Ce sont toujours les mêmes états qui sont obtenus (ce dernier point tient à la structure très simple du modèle dont toutes les fonctions d'effet sont monotones et même linéaires). Remarquons tout d'abord que les résultats respectent la symétrie de la structure du jeu : les satisfactions moyennes des acteurs A2, A3, et A4 sont très proches (85,9 ; 86,1 ; 85,9) ; elles sont bien supérieures à celle de A1 (26,3). A1 n'a pas le choix, il coopère dans tous les cas, ce qui lui vaut sa position au croisement des axes de l'ACP de la figure 4.9. S'il ne coopérait pas, les autres acteurs auraient une satisfaction entre - 80 et - 100, ce qu'ils ne peuvent accepter ; A1 n'est donc pas en mesure d'exercer son pouvoir du fait même de son importance considérable. Il est rare que les trois acteurs A2, A3 et A4 collaborent tous

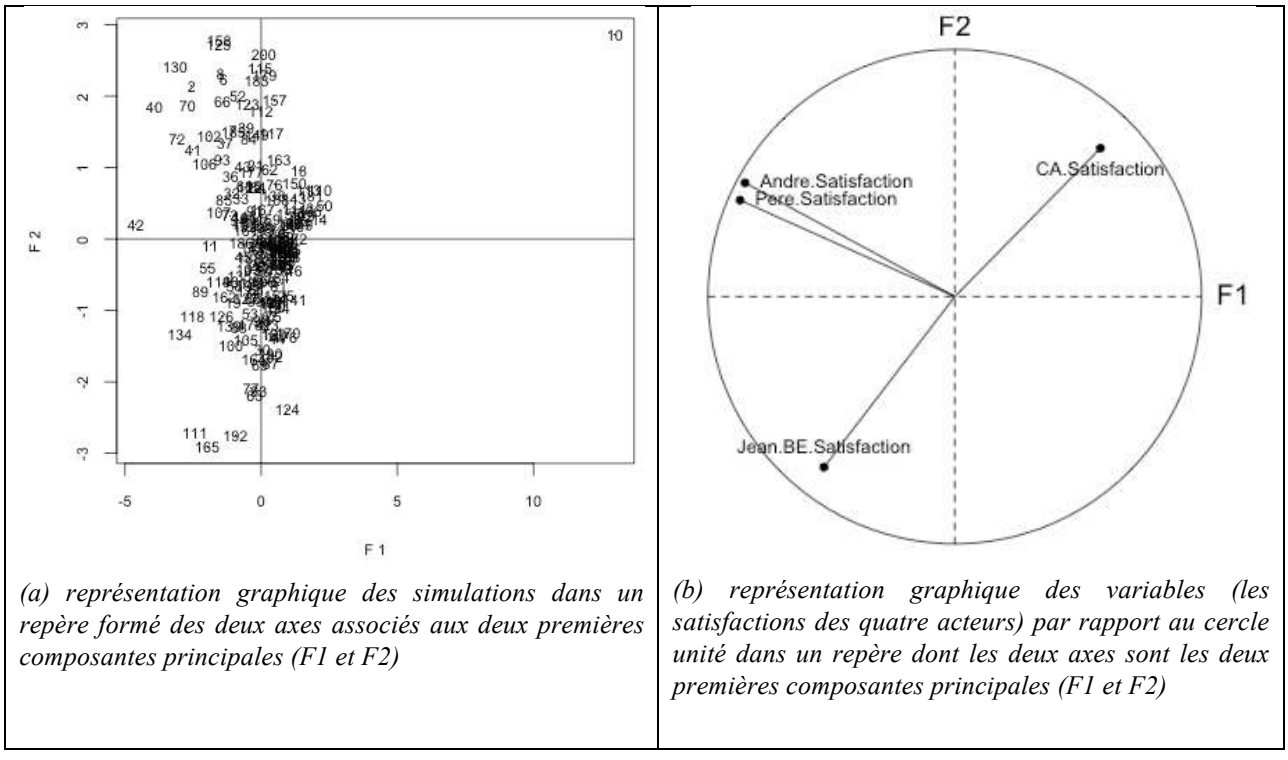
(C1), et on peut remarquer que c'est à l'occasion des simulations courtes, qui se stabilisent en moyenne 5 fois plus rapidement que les autres, comme s'il fallait un certain temps pour que, dans 92,6 % des cas, l'un d'entre eux découvre qu'il peut tirer profit de la coopération des deux autres et faire défection. Enfin, il n'arrive jamais que deux d'entre eux fassent simultanément défection et leur indépendance apparaît clairement ; une telle configuration serait inacceptable pour A1 qui dispose du pouvoir de les en dissuader.

4.4.4 – Le cas Bolet

Le jeu considéré dans cette analyse est celui présenté au paragraphe 2.4.1 du chapitre 2. Les tableaux 4.7 et 4.8 et la figure 4.10 présentent les résultats de 200 simulations. Les paramètres psycho-cognitifs sont initialisés de façon standard : discernement = 1, ténacité = réactivité = 5, et une répartition de 10% pour la dernière règle et 90% pour l'avant dernière règle.

	Satisfaction	Satisfaction en %	Écart-type
CA	-12,03	43,99 %	3,19
Père	56,98	92,85 %	1,08
André	43,61	76,11 %	3,38
Jean-BE	46,97	78,96 %	3,49
Satisfaction Globale	135,53	82,77 %	

Tableau 4.7. *Satisfaction des acteurs à l'issue de 200 simulations du cas Bolet.*



(a) représentation graphique des simulations dans un repère formé des deux axes associés aux deux premières composantes principales (F1 et F2)

(b) représentation graphique des variables (les satisfactions des quatre acteurs) par rapport au cercle unité dans un repère dont les deux axes sont les deux premières composantes principales (F1 et F2)

Figure 4.10. *Analyse en composantes principales des résultats de simulation du cas Bolet obtenue avec le logiciel R (83,15 de la variance est expliquée).*

Le tableau 4.8 caractérise chaque composante principale comme une combinaison linéaire des satisfactions des acteurs. Le premier axe principal est une opposition entre la satisfaction du CA et la satisfaction des trois autres acteurs : le père, André, et jean-BE ; c'est le conflit le plus structurant

du jeu. Tandis que le deuxième axe est une opposition entre la satisfaction de Jean-BE et la satisfaction des trois autres acteurs : CA, André et le père.

	Satisfaction			
	CA	Père	André	Jean-BE
F1	0,59	-0,87	-0,85	-0,53
F2	0,6	0,39	0,46	-0,69

Tableau 4.8. *Contribution des variables (les satisfactions des acteurs) aux deux composantes principales F1 et F2.*

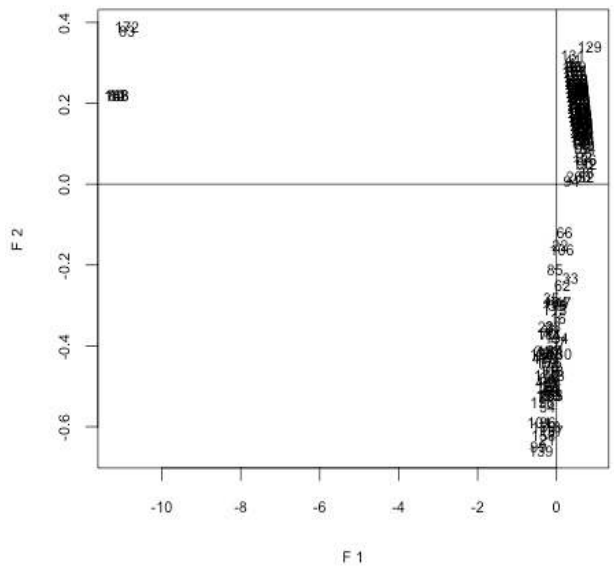
Donnons une interprétation de ces résultats. Cette organisation peut être régulée et donc fonctionner selon deux modalités différentes. La première correspond à une « bonne » satisfaction du chef d'atelier, tandis que la deuxième correspond à une « bonne » satisfaction de Jean-BE. Dans les deux modalités, le Père et André ont une « bonne » satisfaction, bien qu'ils préfèrent s'allier avec Jean-BE : le tableau 4.8 montre une très forte opposition entre la satisfaction du CA et les satisfactions du père et André pour F1, et une opposition moindre entre la satisfaction du Jean-BE et les satisfactions du père et André pour F2. Une analyse de sensibilité sur la ténacité du chef d'atelier, dans [El Gemayel *et al.*, 2011], a montré que plus le chef d'atelier est tenace, plus il arrive à convaincre le père et André de le rejoindre contre Jean-BE, et plus il améliore sa satisfaction.

4.4.5 – Le cas Seita

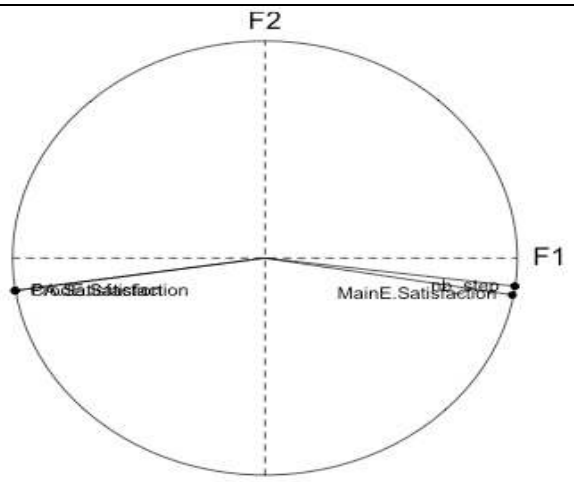
Le jeu considéré dans cette analyse est celui présenté au paragraphe 2.4.2 du chapitre 2. Les tableaux 4.9 et 4.10 et la figure 4.11 présentent les résultats de 200 simulations. Les paramètres psycho-cognitifs sont initialisés de façon standard : discernement = 1, ténacité = réactivité = 5, et une répartition de 30% (ou 20% ou 50% en fonction de la somme des enjeux posés sur la relation que l'acteur contrôle) pour la dernière règle et 70% (ou 80% ou 50%) pour l'avant dernière règle.

	Satisfaction	Satisfaction en %	Écart-type
CA	-52,15	17,26 %	8,1
ProdE	-4,67	47,26 %	7,87
MainE	79,43	98,16 %	3,92
Satisfaction Globale	22,61	28,96 %	

Tableau 4.9. *Satisfaction des acteurs à l'issue de 200 simulations du cas Seita.*



(a) représentation graphique des simulations dans un repère formé des deux axes associés aux deux premières composantes principales (F1 et F2)



(b) représentation graphique des variables (les satisfactions des quatre acteurs et le nombre de pas) par rapport au cercle unité dans un repère dont les deux axes sont les deux premières composantes principales (F1 et F2) ; les satisfactions du CA et ProdE sont à gauche, tandis que la satisfaction du MainE et le nb_step sont à droite.

Figure 4.11. *Analyse en composantes principales des résultats de simulation du cas Seita obtenue avec le logiciel R (99,66 de la variance est expliquée, dont 97,39 est expliqué par le première composante F1).*

Le tableau 4.10 caractérise chaque composante principale comme une combinaison linéaire des satisfactions des acteurs. Le premier Axe principal représente une très forte opposition entre les satisfactions du chef d'atelier et des ouvriers de production, et la satisfaction des ouvriers de maintenance et le nombre de pas pour la convergence. Tandis que le deuxième axe n'est pas significatif, puisque 97% de la variance est expliquée par le premier axe.

		Satisfactions		
	Nombre de pas	CA	ProdE	MainE
F1	0,99	-0,99	-0,99	0,98
F2	-0,13	-0,15	-0,15	-0,17

Tableau 4.10. *Contribution des variables (les satisfactions des acteurs et le nombre de pas) aux deux composantes principales F1 et F2.*

Ces résultats montrent un conflit structurel entre, d'une part le chef d'atelier et les ouvriers de production et, d'autre part les ouvriers de maintenance. La majorité des simulations sont courtes et au bénéfice des ouvriers de maintenance. Le chef d'atelier arrive rarement à augmenter sa satisfaction quand les simulations durent longtemps. En effet, les ouvriers de maintenance posent 5 points d'enjeux sur les relations qu'ils contrôlent et 5 point d'enjeux sur les deux autres relations ; ils contrôlent à moitié la réalisation de leurs objectifs, et ils n'ont aucun intérêt à coopérer avec le chef d'atelier tant que cette coopération est au bénéfice exclusive du chef d'atelier. De plus, la relation « pression », contrôlée par les ouvriers de maintenance, contraint la relation « règle », contrôlée par le chef d'atelier : plus les ouvriers de maintenance sont agressifs envers le chef d'atelier, plus l'espace de décision du chef d'atelier est restreint, et plus il est forcé à appliquer les règles à la lettre. De même, la relation « maintenance », contrôlée par les ouvriers de maintenance, contraint la relation « production », contrôlée par les ouvriers de production : plus la maintenance

des machines est faite de façon imprévisible, plus la production diminue. Ce qui rend le chef d'atelier et les ouvriers de production incapables d'appliquer leur pouvoir.

4.4.6 – Le cas Touch

Le jeu considéré dans cette analyse est celui présenté au paragraphe 2.4.3 du chapitre 2. Le tableau 4.11 présente les résultats de 200 simulations. Les paramètres psycho-cognitifs sont initialisés de façon standard : discernement = 1, ténacité = réactivité = 5, et une répartition de 40% (ou 30% en fonction de la somme des enjeux posés sur la relation que l'acteur contrôle) pour la dernière règle et 60% (ou 70%) pour l'avant dernière règle.

	Satisfaction	Satisfaction en %	Écart-type
DDT	57,21	94,25 %	0,91
ONEMA	53,63	84,15 %	0,9
Agence de l'eau	64,01	86,66 %	0,63
ARTESA	55,72	82,65 %	0,46
Communes_amont	49,67	81,40 %	1,68
Communes_aval	35,59	59,81 %	2,28
SIAH	55,86	88,72 %	1,6
CR	53,57	93,38 %	1,3
CG	54,06	88,99 %	0,68
SOGREAH	47,91	76,33 %	1,94
Satisfaction Globale	527,23	94,41 %	

Tableau 4.11. *Satisfaction des acteurs à l'issue de 100 simulations du cas Touch.*

Malgré l'écart entre les satisfactions des acteurs (certains ont atteint 90% de leur satisfaction maximale, d'autres environ 70%), la valeur de la satisfaction globale, qui est la somme des satisfactions des acteurs, est très élevée, et proche du maximum. Ce qui valide le fait que notre algorithme produit des résultats proches des optima de Pareto : les acteurs sociaux coopèrent tant que la coopération est profitable, et chacun cherche une solution satisfaisante, mais pas nécessairement optimale pour lui. Les acteurs les moins satisfaits en pourcentage sont ceux dont l'écart type est le plus important.

4.5 – Analyse de sensibilité des paramètres psycho-cognitifs

L'analyse de sensibilité permet d'analyser l'influence de paramètres d'entrée sur les résultats de l'algorithme. Cette analyse peut être appliquée sur un ou plusieurs paramètres d'entrée en même temps, mais plus le nombre des paramètres d'entrée augmente, plus l'interprétation et l'explication des résultats sont difficiles à établir. Elle consiste à mettre en évidence :

- Quels sont les paramètres d'entrée les plus influents sur la valeur et éventuellement la variation des variables de sortie ?
- Dans quel(s) cas certains paramètres sont moins influents que dans d'autres cas ?
- Et quelles sont les valeurs optimales des paramètres d'entrée, c'est-à-dire qui permettent le mieux d'obtenir les résultats attendus ?

Il s'agit donc d'une analyse à la fois qualitative et quantitative. Des analyses de sensibilité de chacun des paramètres psycho-cognitifs sont faites sur plusieurs modèles d'organisations virtuelles et réelles. Dans cette section, nous n'en présentons que quelques-unes qui mettent en évidence l'importance et l'effet du paramètre étudié sur les résultats de simulation. Ensuite nous concluons sur les résultats obtenus.

4.5.1 – Analyse de la ténacité

La ténacité d'un acteur détermine sa pugnacité, à quel point il essaie d'obtenir des autres la plus grande capacité d'atteindre ses objectifs ; elle est le contraire de la résignation. Dans [El Gemayel *et al.*, 2011], nous avons analysé l'impact de la ténacité sur le comportement des acteurs sociaux dans deux modèles d'organisation : le dilemme du prisonnier et le cas Bolet. Dans le dilemme du prisonnier, il apparaît que plus les acteurs sont tenaces, plus les simulations vont durer, et plus les acteurs coopèrent même quand l'intérêt à le faire est faible. Dans le cas Bolet, augmenter la ténacité de tous ne changeait rien aux résultats, par contre augmenter la ténacité du chef d'atelier accroit sa satisfaction et décroit celle de Jean-BE. Ainsi, la valeur appropriée de la ténacité des acteurs est celle qui permet d'obtenir les résultats à un coût raisonnable (nombre de pas nécessaires pour la convergence) tout en assurant la qualité de coopération attendue ou recherchée (la satisfaction finale des acteurs). En tout état de cause, cette valeur dépend de propriétés structurelles de l'organisation qui déterminent la difficulté, pour chaque acteur, de trouver comment coopérer avec les autres.

Pour étudier l'influence de la ténacité sur les résultats des simulations, notamment sur le nombre de pas nécessaires pour la convergence et le niveau de satisfaction acquise par les acteurs, nous analysons les résultats de l'algorithme de simulation sur deux modèles d'organisation : le dilemme du prisonnier, et le modèle free rider.

Le dilemme du prisonnier 4/6

Nous considérons ici un dilemme du prisonnier avec une répartition des enjeux 4/6. Cette répartition représente un cas spécial : les acteurs ont intérêt à coopérer pour obtenir une meilleure situation, mais ils n'arrivent pas à détecter l'importance de la coopération durant la simulation (cf. 4.4.1) à cause de leur faible ténacité et du faible bénéfice de la coopération.

Les figures 4.12 et 4.13 montrent les résultats d'une analyse de sensibilité, comprenant 10 expériences où la ténacité est la même pour les deux acteurs et varie entre 1 et 10. Chaque expérience de l'analyse de sensibilité a été réalisée avec 100 simulations. Les paramètres psycho-cognitifs sont initialisés de façon standard : discernement = 1, réactivité = 5, et une répartition de 40% pour la dernière règle et 60% pour l'avant dernière règle.

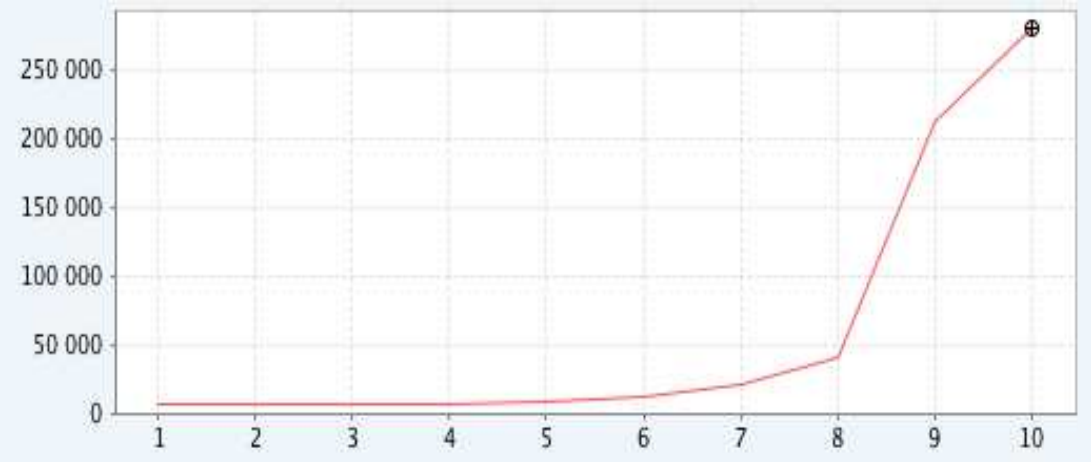


Figure 4.12. *La moyenne du nombre de pas nécessaires pour la convergence, en fonction de la ténacité des deux acteurs.*

Il semble à l'examen de la figure 4.12 que le nombre de pas nécessaires pour la convergence augmente de façon quasi exponentielle avec l'augmentation de la ténacité des deux acteurs : pour une ténacité de 10 pour les deux acteurs, il n'y a que 4% des simulations qui convergent en moins de 300 000 pas. Le nombre de pas varie légèrement d'environ 5 800 à 8 100 quand la ténacité augmente de 1 à 5, tandis qu'il augmente brusquement d'environ 11 500 à 290 000 quand la ténacité augmente de 6 à 10.

Les satisfactions des deux acteurs (cf. figure 4.13) sont semblables quelque soit la valeur de la ténacité. Elles sont quasiment constantes à -12 quand la ténacité augmente de 1 à 4, augmentent jusqu'à 18 quand la ténacité augmente à 8, et se stabilisent à 20 pour une ténacité de 9 et 10.

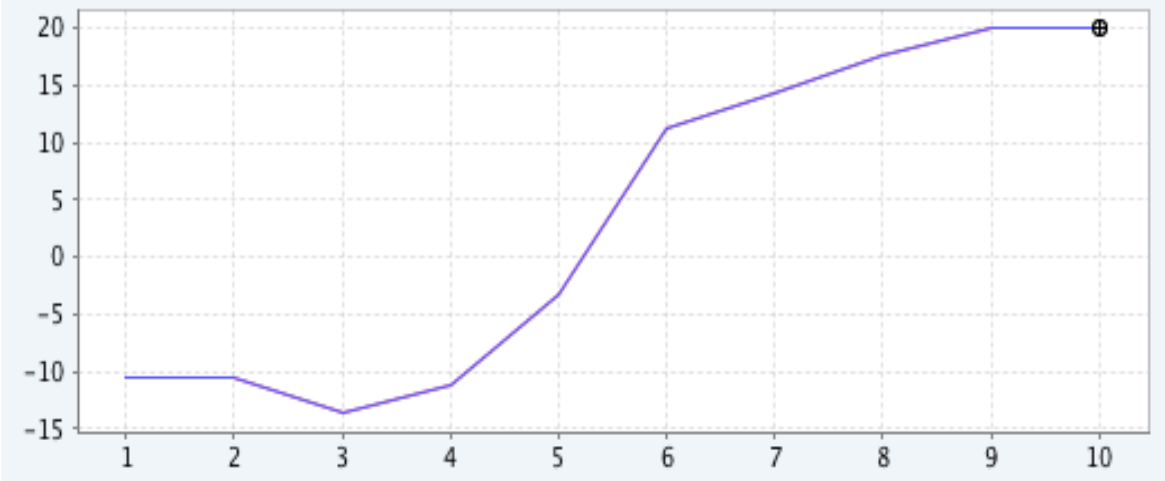


Figure 4.13. *La moyenne de la satisfaction de chaque acteur, en fonction de la ténacité des deux acteurs.*

Ces résultats peuvent être expliqués : un acteur peu tenace se contente d'une situation peu satisfaisante et ne cherche pas à l'améliorer. Plus un acteur est tenace, plus il va chercher à améliorer sa situation. Au delà d'un certain seuil, 8 pour cet exemple (cf. figure 4.12), l'acteur va explorer presque tout le temps quelque soit son écart, et les simulations deviennent très coûteuses.

Nous avons aussi considéré le cas où la ténacité de l'acteur A1 varie entre 1 et 10, tandis que la ténacité de l'autre est fixée à 5. Le nombre de pas varie presque de la même façon (cf. figure 4.14) : il est quasiment constant à 10 500 quand la ténacité de A1 augmente de 1 à 5, puis augmente à 45 000 quand la ténacité augmente à 9, et est à 42 500 pour une ténacité de 10. Les satisfactions des deux acteurs sont semblables et varient de façon parabolique (cf. figure 4.15): elles sont inférieures à -13,5 pour une ténacité de 1 à 4, augmentent à -4 quand la ténacité de A1 augmente de 4 à 6, puis subissent une chute de -3 à environ -17 quand la ténacité augmente de 6 à 8, et se stabilisent à environ -16 pour une ténacité de 9 ou 10. Les satisfactions les plus élevées sont obtenues pour une ténacité de 5 ou 6 de A1 quasiment égale à celle de A2, tandis que pour les autres valeurs de ténacité les satisfactions sont mauvaises. En effet, lorsqu'un acteur est plus tenace que l'autre, il explore plus pour améliorer sa satisfaction, tandis que l'autre acteur, le moins tenace, se contente d'obtenir une « bonne » satisfaction sans chercher à l'améliorer, et joue défensivement en préservant l'impact de la relation qu'il contrôle afin d'éviter le pire.

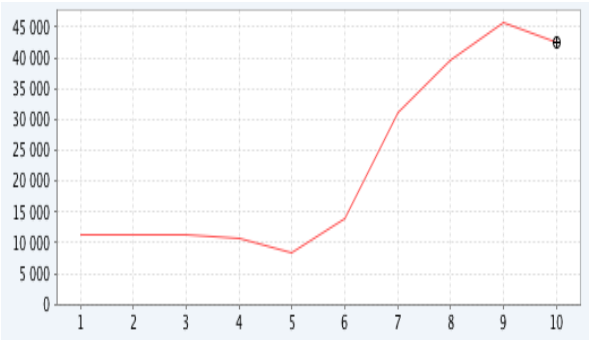


Figure 4.14. *La moyenne du nombre de pas nécessaires pour la convergence, en fonction de la ténacité de A1.*

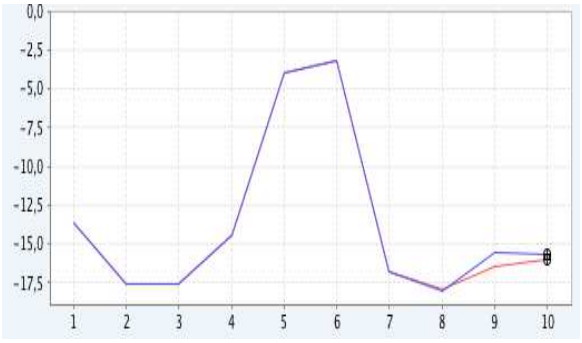


Figure 4.15. *La moyenne de la satisfaction de chaque acteur, en fonction de la ténacité de A1.*

Le modèle free-rider

Les figures 4.16 et 4.17 montrent les résultats d'une analyse de sensibilité, comprenant 10 expériences où la ténacité de l'acteur Act1 varie entre 1 et 10, tandis que la ténacité des trois autres est fixée à 5. Chaque expérience d'analyse de sensibilité a été réalisée avec 100 simulations. Les paramètres psycho-cognitifs sont initialisés de façon standard : discernement = 1, réactivité = 5, et une répartition de 10% pour la dernière règle et 90% pour l'avant dernière règle.

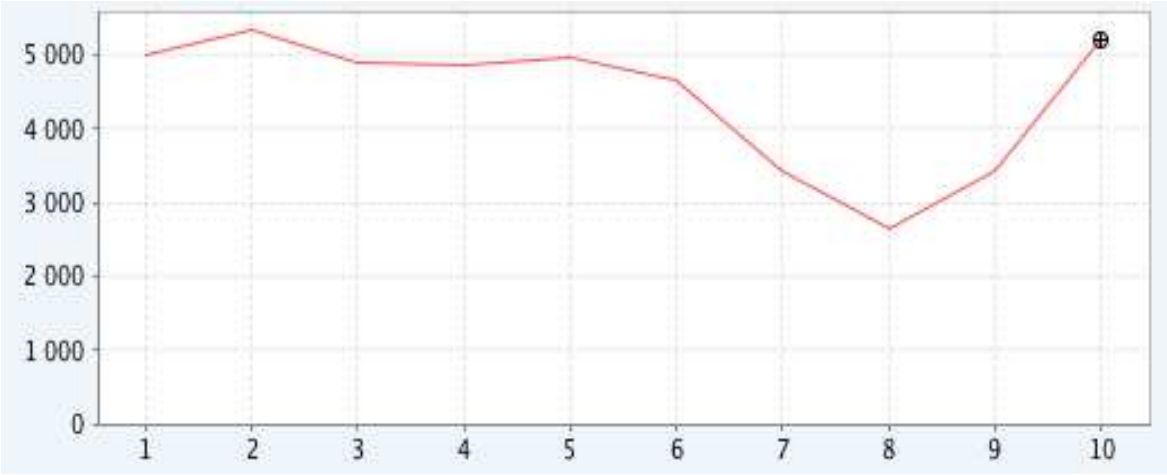


Figure 4.16. *La moyenne des nombres de pas nécessaires pour la convergence, en fonction de la ténacité de l'acteur Act1.*

Le nombre de pas nécessaires pour la convergence est quasiment constant à 5 000 sauf pour une ténacité de l'acteur Act1 entre 7 et 9 : il subit une chute jusqu'à environ 3 400, puis 2800, et il remonte à 3 400 (cf. figure 4.16).

Les satisfactions des trois acteurs (Act2, Act3, et Act4) diminuent légèrement d'environ 88 à 80 quand la ténacité de l'acteur Act1 est améliorée, tandis que la satisfaction de l'acteur Act1 augmente rapidement jusqu'à atteindre une satisfaction sociale maximale pour une ténacité entre 8 et 10 (cf. figure 4.17).

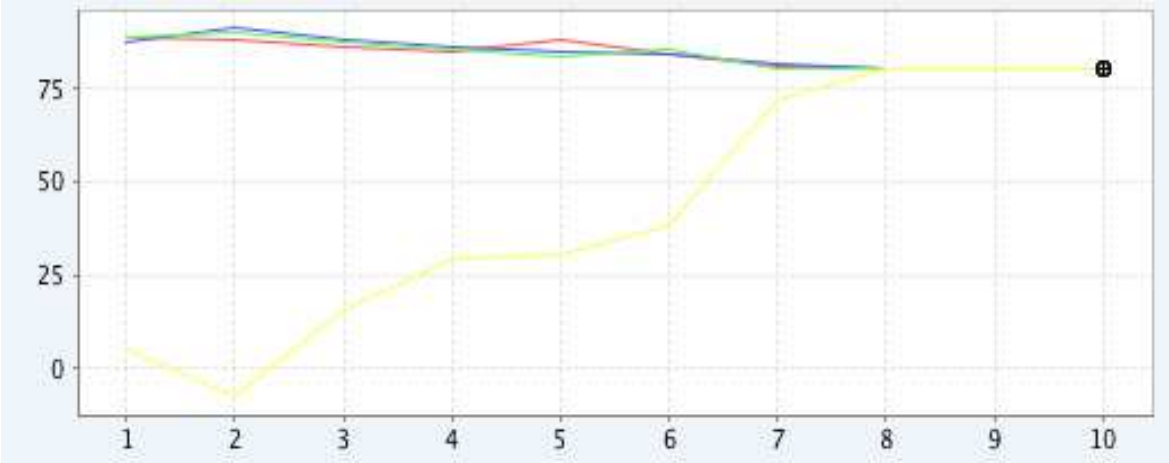


Figure 4.17. *La moyenne de la satisfaction de chaque acteur (Act1 Jaune, Act2 Vert, Act3 Bleu, Act4 Rouge), en fonction de la ténacité de l'acteur Act1.*

Il semble que pour ce jeu, la ténacité de l'acteur Act1 n'a pas beaucoup d'influence sur le nombre de pas nécessaires pour la convergence. Par contre, il apparaît clairement qu'elle influence la satisfaction des acteurs, notamment la satisfaction de l'acteur Act1. Une ténacité de 8 pour l'acteur Act1 semble la meilleure valeur vis-à-vis les résultats : une simulation peu coûteuse et le jeu se stabilise dans l'état qui maximise la satisfaction globale (la configuration C1, cf. tableau 4.5)

Dans le cas où la ténacité de tous les acteurs varie entre 1 et 10, les résultats sont comparables à ceux présentés ci-dessus (cf. figure 4.18 et 4.19) : les satisfactions des acteurs (Act2, Act3, et Act4) diminuent légèrement de 89 à 80, tandis que la satisfaction de l'acteur Act1 augmente rapidement de -8 jusqu'à atteindre sa satisfaction sociale maximale pour une ténacité entre 8 et 10. Le nombre de pas est quasiment constant à 3 000 pour une ténacité entre 1 et 8, et augmente brusquement à 18 000 pour une ténacité de 10.

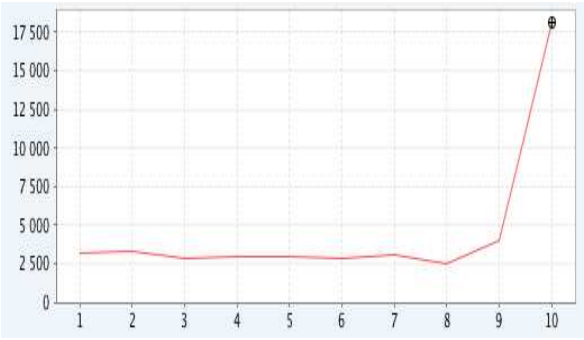


Figure 4.18. *La moyenne des nombres de pas nécessaires pour la convergence, en fonction de la ténacité de tous les acteurs.*

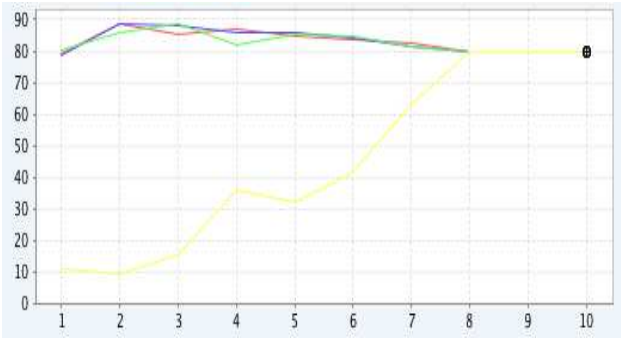


Figure 4.19. *La moyenne de la satisfaction de chaque acteur (Act1 Jaune, Act2 Vert, Act3 Bleu, Act4 Rouge), en fonction de la ténacité de tous les acteurs.*

Discussion

La ténacité a été introduite dans l'algorithme de simulation pour déterminer le niveau d'exploration par rapport au niveau d'exploitation. Elle intervient dans la mise à jour du taux d'exploration de l'acteur, plus précisément dans l'équation du taux d'exploration instantanée.

$$TXI_t = 0.1 + \left(\frac{0.8}{1 + e^{pente \times (écart_t - abscisse)}}\right)$$

où *pente* = - (*ténacité* × (10 - *ténacité*) + 10) et *abscisse* = (10 - *ténacité*) / 10

Dans l'ensemble, l'augmentation de la ténacité produit un prolongement de l'exploration qui à son tour produit, dans la plupart des cas, une augmentation de la durée des simulations, et cette augmentation est de plus en plus importante pour une ténacité proche de sa valeur maximale (entre 8 et 10). Cependant, les choses ne se passent pas toujours ainsi, par exemple le modèle free-rider, où une ténacité entre 6 et 9 produit une diminution de la durée des simulations. Dans le cas de ce modèle, lorsque les simulations durent longtemps, c'est du fait de l'acteur A1 dont la satisfaction reste très au-dessous de son ambition (initialisée à 100), alors que les autres acteurs obtiennent rapidement un haut niveau de satisfaction. En explorant davantage avec une ténacité élevée, A1 parvient à convaincre les trois autres de coopérer et sa satisfaction augmente suffisamment pour que la simulation converge. Par contre, s'il est trop exigeant avec une ténacité de 9 ou 10, la simulation se prolonge, sans pouvoir améliorer sa satisfaction puisque le jeu est déjà dans l'état correspondant à son optimum global.

L'augmentation de la ténacité produit, dans la plupart des cas, une augmentation significative des satisfactions des acteurs. Dans certaines organisations, où la coopération est imposée par la structure de l'organisation elle-même (par exemple le dilemme du prisonnier classique avec une répartition 1 / 9), la ténacité n'a pas beaucoup d'effet sur les satisfactions des acteurs ; dans ces cas, l'acteur n'a pas le choix, il doit coopérer quelque soit sa ténacité, car cette coopération lui est extrêmement profitable. Dans d'autres organisations, par exemple le dilemme du prisonnier classique avec une répartition 4 / 6 où la ténacité d'un acteur est plus importante que celle de l'autre, celui le moins tenace joue de façon défensive en préservant sa propre relation pour lui-même, ce qui force l'acteur le plus tenace à faire pareil, et le jeu se stabilise dans un équilibre de Nash.

Dans tous les cas, la valeur du paramètre ténacité choisie par le modélisateur doit être un compromis entre les deux effets de l'augmentation de la ténacité : une simulation plus coûteuse en temps de calcul, et une coopération plus élevée. Pour ce faire, on peut considérer la fonction suivante (cf. éq. 4.18) qui mesure, en proportion, l'écart entre la satisfaction d'un acteur et la durée des simulations par rapport à leurs valeurs maximales et minimales :

$$InfluenceTénacité = \left(\frac{satisfaction - satisfactionMin}{satisfactionMax - satisfactionMin} \right) - \left(\frac{nbEtapes - nbEtapesMin}{nbEtapesMax - nbEtapesMin} \right) \quad \text{(éq. 4.18)}$$

où satisfactionMax, satisfactionMin, nbEtapesMax, et nbEtapesMin sont les valeurs maximales et minimales obtenues à la fin des simulations. Cette fonction est sans unité et évite de comparer les valeurs elles-mêmes, dont les domaines sont très différents. L'augmentation de la ténacité au-delà de la valeur correspondant au maximum de cette fonction produit une prolongation des simulations qui, en proportion, est plus importante que l'amélioration de la satisfaction. Une « bonne » ténacité est celle pour laquelle cette fonction dépasse un certain seuil (par exemple fixé à 0,5, dans le modèle free-rider une bonne ténacité est alors entre 7 et 9, cf. figure 4.20), tandis que la meilleur ténacité est celle pour laquelle cette fonction est maximale (pour une ténacité de 8 pour le modèle free-rider, cf. figure 4.20).

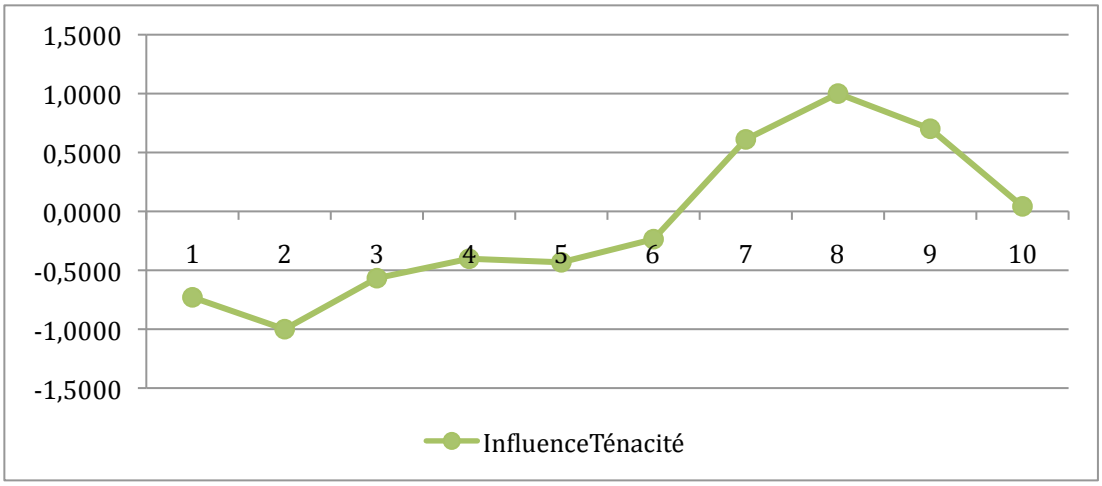


Figure 4.20. *La courbe de l'InfluenceTénacité en fonction de la ténacité de l'acteur A1 dans le cas free-rider.*

4.5.2 – Analyse de la réactivité

La réactivité d'un acteur détermine la part relative de l'expérience et des nouvelles informations dans son apprentissage, l'importance qu'il attache à ce qu'il sait déjà dans la mise à jour de son taux d'exploration et de son ambition. Il apparaît que plus les acteurs sont réactifs, plus les simulations sont courtes, sans trop affecter la qualité de la coopération. Ainsi, la valeur appropriée de la réactivité des acteurs est un compromis entre le coût des simulations et la qualité de la coopération. Il semble également que cette valeur dépende de la structure de l'organisation, surtout de la possibilité de prévoir le comportement des autres acteurs.

Pour étudier l'influence de la réactivité sur le nombre de pas nécessaires pour la convergence et le niveau de satisfaction acquise par les acteurs, nous analysons les résultats de l'algorithme de simulation sur deux types de modèles d'organisation : le dilemme du prisonnier classique et le modèle free-rider.

Le dilemme du prisonnier 4/6

Le dilemme du prisonnier traité dans cette analyse est celui avec une répartition 4 / 6. Les figures 4.21 et 4.22 montrent les résultats d'une analyse de sensibilité du dilemme du prisonnier, comprenant 10 expériences où la réactivité est la même pour les deux acteurs et varie entre 1 et 10. Chaque expérience d'analyse de sensibilité a été réalisée avec 100 simulations. Les paramètres psycho-cognitifs sont initialisés de façon standard : discernement = 1, ténacité = 5, et une répartition de 40% pour la dernière règle et 60% pour l'avant dernière règle.

Figure 4.21. *La moyenne des nombres de pas nécessaires pour la convergence, en fonction de la réactivité des deux acteurs.*

Il est clair à l'examen de la figure 4.21 que le nombre de pas nécessaires pour la convergence diminue de façon exponentielle avec l'augmentation de la réactivité des deux acteurs. Il diminue brusquement d'environ 33 500 à 10 000 quand la réactivité augmente de 1 à 4, tandis qu'il diminue légèrement d'environ 10 000 à 4 700 quand la ténacité augmente de 4 à 10.

Les satisfactions des deux acteurs (cf. figure 4.22) sont semblables quelque soit la valeur de la réactivité. Elles diminuent de façon quasi linéaire d'environ 17 à -12 quand la réactivité augmente de 1 à 10, avec une satisfaction de 2,4 pour une réactivité de 4.

Figure 4.22. *La moyenne de la satisfaction de chaque acteur, en fonction de la réactivité des deux acteurs.*

Ces résultats peuvent être expliqués : un acteur trop réactif semble être naïf et ne pas prendre le temps de bien évaluer la réaction des autres acteurs. Moins un acteur est réactif, mieux il évalue la réaction des autres, plus il améliore sa situation.

Nous avons aussi considéré le cas où la réactivité d'un seul des deux acteurs, A1, varie entre 1 et 10, tandis que la réactivité de l'autre est fixée à 5. Les résultats sont globalement similaires au cas précédent. Le nombre de pas diminue brusquement d'environ 46 000 à 11 200 quand la réactivité de l'acteur A1 augmente de 1 à 4, puis diminue légèrement jusqu'à 8 700 quand la réactivité augmente à 10 (cf. figure 4.23). Les satisfactions des deux acteurs sont semblables, et varient légèrement de façon aléatoire entre -6 et 0 quand la réactivité de A1 augmente de 1 à 10 ; une satisfaction maximale de 0 est obtenue pour une réactivité de 5, similaire à la réactivité de l'autre acteur (cf. figure 4.24). En outre, lorsqu'un acteur est plus réactif que l'autre, son taux d'exploration varie plus vite que celui de l'autre acteur, et suit de plus prêt sa satisfaction. L'autre acteur, étant moins réactif, cherche à s'assurer de la stabilité de sa nouvelle situation avant de la prendre en compte, ce qui semble très difficile étant donné que sa situation dépend du comportement imprévisible de l'autre acteur. Ce décalage entre les vitesses avec lesquelles chaque acteur met à jour son taux d'exploration rend les résultats de simulation peu aléatoire.

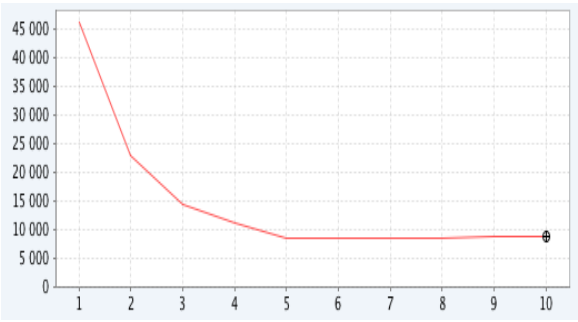


Figure 4.23. *La moyenne du nombre de pas nécessaires pour la convergence, en fonction de la réactivité de A1.*

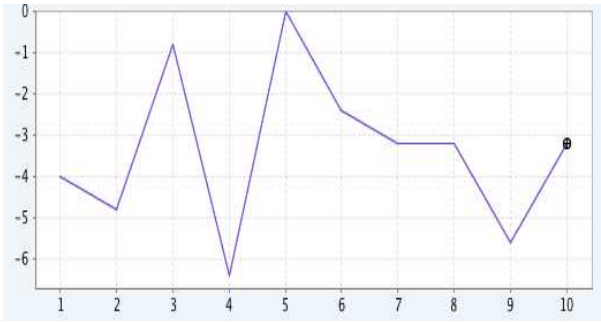


Figure 4.24. *La moyenne de la satisfaction de chaque acteur, en fonction de la réactivité de A1.*

Le modèle free-rider

Les figures 4.25 et 4.26 montrent les résultats d'une analyse de sensibilité du modèle free rider, comprenant 10 expériences, où la réactivité de l'acteur Act1 varie entre 1 et 10, tandis que la réactivité des trois autres est fixée à 5. Chaque expérience d'analyse de sensibilité a été réalisée avec 100 simulations. Les paramètres psycho-cognitifs sont initialisés de façon standard : discernement = 1, ténacité = 5, et une répartition de 10% pour la dernière règle et 90% pour l'avant dernière règle.

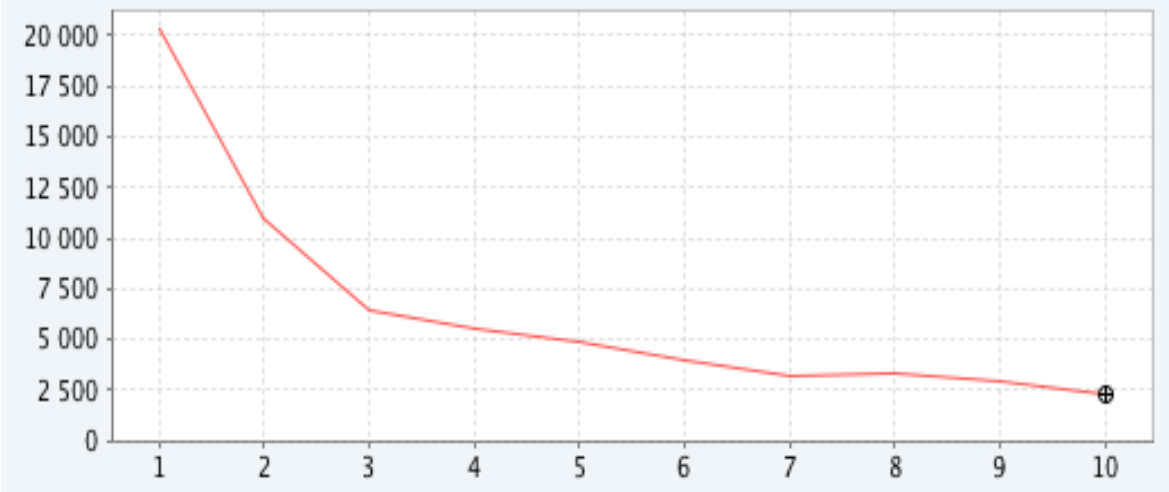


Figure 4.25. *La moyenne des nombres de pas nécessaires pour la convergence, en fonction de la ténacité de l'acteur Act1.*

Le nombre de pas nécessaires pour la convergence diminue exponentiellement d'environ 20 000 à 2 500 avec l'augmentation de la réactivité de l'acteur Act1 (cf. figure 4.25), selon la même forme générale que le dilemme du prisonnier.

La satisfaction de l'acteur A1 varie de façon aléatoire avec l'augmentation de sa réactivité, tandis que les satisfactions des trois acteurs (Act2, Act3, et Act4) varient peu à l'opposé de la variation de la satisfaction de l'acteur A1 (cf. figure 4.26). Cette différence entre les variations de satisfaction de Act1 et des autres acteurs s'applique par le fait que ces derniers ne mettent qu'un seul point d'enjeu sur les relations dont Act1 dépend.

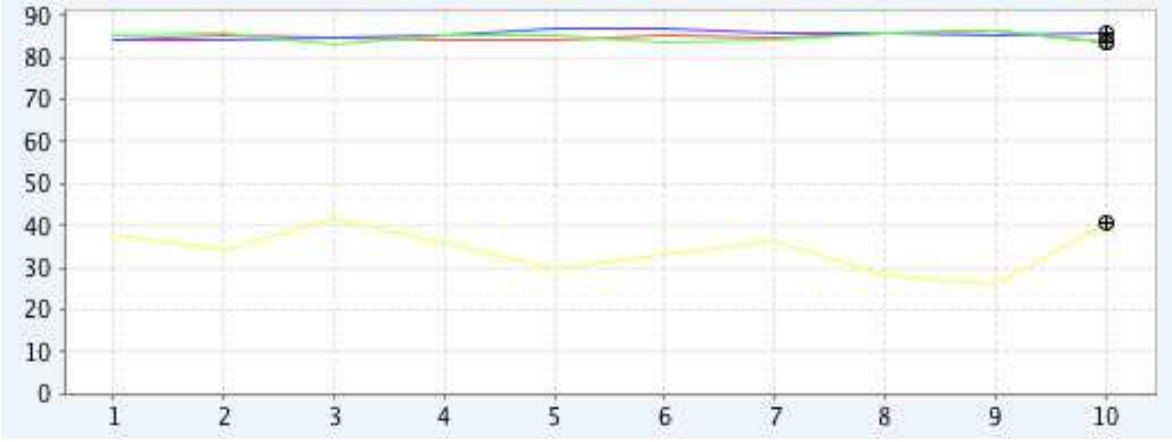


Figure 4.26. *La moyenne de la satisfaction de chaque acteur (Act1 Jaune, Act2 Vert, Act3 Bleu, Act4 Rouge), en fonction de la ténacité de l'acteur Act1.*

Il semble que pour ce jeu, la réactivité de l'acteur Act1 n'a pas d'influence sur les satisfactions des acteurs. Par contre, il apparaît clairement qu'elle influence le nombre de pas

nécessaires pour la convergence. Une réactivité maximale de 10 pour l'acteur Act1 diminue la durée des simulations sans obérer les résultats.

Nous avons aussi considéré le cas où tous les acteurs ont la même réactivité qui varie entre 1 et 10. Les résultats sont comparables au cas précédent où seul la réactivité de A1 varie : les satisfactions des trois acteurs (Act2, Act3, et Act4) varient légèrement de façon aléatoire, tandis que la satisfaction de l'acteur Act1 varie à l'opposé de la variation de la satisfaction des autres acteurs (cf. figure 4.28). Le nombre de pas nécessaires pour la convergence diminue d'environ 20 500 à 2 500 avec l'augmentation de la réactivité des acteurs (cf. figure 4.27).

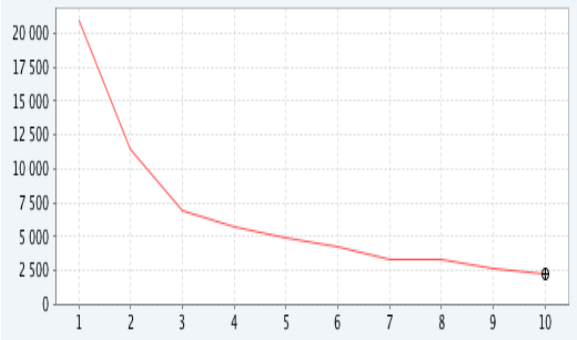


Figure 4.27. *La moyenne des nombres de pas nécessaires pour la convergence, en fonction de la réactivité de tous les acteurs.*

Figure 4.28. *La moyenne de la satisfaction de chaque acteur (Act1 Jaune, Act2 Vert, Act3 Bleu, Act4 Rouge), en fonction de la réactivité de tous les acteurs.*

Discussion

La réactivité a été introduite dans l'algorithme de simulation pour déterminer à quel point un acteur attache de l'importance au passé et avec quelle vitesse il fait varier, notamment diminuer, son ambition : plus la réactivité diminue, plus le poids du passé augmente, et plus l'ambition de l'acteur varie lentement. Elle intervient dans la mise à jour du taux d'exploration de l'acteur et de son ambition.

$$ambition_t = ambition_{t-1} - ((1 - TX_{t-1}) \times (réactivité/100) \times écart_t)$$

$$ambition_t = ambition_{t-1} + ((capacitéAction_t - ambition_{t-1}) \times (réactivité/100))$$

$$TX_t = (1 - réactivité) \times TX_{t-1} + réactivité \times TXI_t$$

La diminution de la réactivité d'un acteur produit un ralentissement de la variation de son ambition qui à son tour produit une augmentation de la durée des simulations, et cette augmentation est de plus en plus importante pour une réactivité proche de la valeur minimale (entre 1 et 3).

Par ailleurs, la diminution de la réactivité produit, dans certains cas, par exemple le dilemme du prisonnier avec une répartition 4 / 6, une augmentation des satisfactions des acteurs. En effet, un acteur très réactif suit de prêt la réaction des autres acteurs, et réagit rapidement sans s'assurer de la bonne coopération : il peut arriver qu'un acteur A prenne une décision aléatoire, bonne pour l'acteur B, sans avoir l'intention de coopérer avec lui. L'acteur B, étant très réactif, réagit en coopérant avec l'acteur A. Ce dernier peut ainsi profiter de la coopération de l'acteur B, et jouer la trahison. Par contre, un acteur peu réactif arrive à mieux analyser les réactions des autres acteurs, et à s'assurer de la bonne coopération avant de coopérer à son tour avec les autres.

L'influence de la réactivité sur le déroulement des simulations semble assez similaire, en inversé, à celle de la ténacité. Dans tous les cas, la valeur du paramètre réactivité choisie par le modélisateur doit être un compromis entre les deux effets de la diminution de la réactivité : une simulation plus coûteuse en temps de calcul, et une coopération plus élevée. Pour ce faire, on peut considérer la même fonction que pour la ténacité (cf. éq. 4.19), qui mesure, en proportion, l'écart

entre la satisfaction d'un acteur et la durée des simulations par rapport à leurs valeurs maximales et minimales :

$$InfluenceRéactivité = \left(\frac{satisfaction - satisfactionMin}{satisfactionMax - satisfactionMin} \right) - \left(\frac{nbEtapes - nbEtapesMin}{nbEtapesMax - nbEtapesMin} \right) \quad \text{(éq. 4.19)}$$

où satisfactionMax, satisfactionMin, nbEtapesMax, et nbEtapesMin sont les valeurs maximales et minimales obtenues à la fin des simulations. Une « bonne » réactivité est celle pour laquelle cette fonction dépasse un certain seuil (par exemple fixé à 0,3, dans le cas du dilemme du prisonnier une bonne réactivité est alors de 2 ou 4, cf. figure 4.29), tandis que la meilleur réactivité est celle pour laquelle cette fonction est maximale (pour une réactivité de 2 dans le cas du dilemme du prisonnier, cf. figure 4.29).

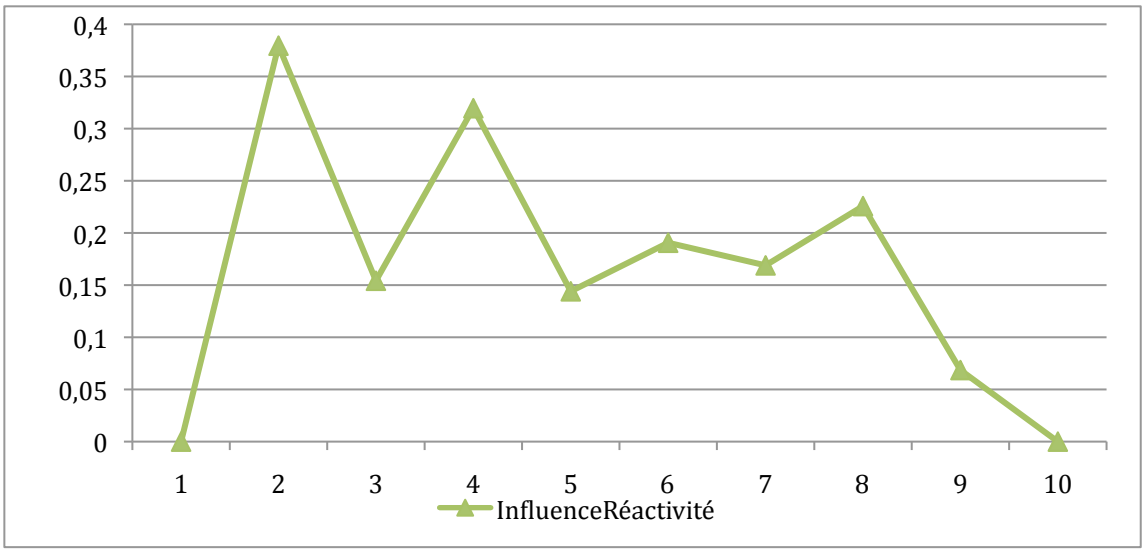


Figure 4.29. La courbe de l'InfluenceRéactivité en fonction de la réactivité des deux acteurs dans le cas du dilemme du prisonnier.

4.5.3 – La ténacité & la réactivité

La ténacité et la réactivité d'un acteur semblent avoir beaucoup d'influence sur les résultats de simulation. Avant de procéder aux analyses de sensibilité des autres paramètres psycho-cognitifs, une analyse de sensibilité combinée de l'influence de ces deux paramètres sur les résultats des simulations est nécessaire afin de trouver la meilleure combinaison de valeurs. Pour ce faire, nous analysons les résultats de l'algorithme sur deux modèles d'organisation : le dilemme du prisonnier et le modèle free-rider.

Le dilemme du prisonnier 4/6

Le dilemme du prisonnier traité dans cette analyse est à nouveau celui avec une répartition 4 / 6. Les figures 4.30 et 4.31 montrent les résultats d'une analyse de sensibilité, comprenant 100 expériences où les deux acteurs ont les mêmes valeurs de ténacité et réactivité qui varient entre 1 et 10. Chaque expérience d'analyse de sensibilité a été réalisée avec 100 simulations. Les paramètres psycho-cognitifs sont initialisés de façon standard : discernement = 1, et une répartition de 40% pour la dernière règle et 60% pour l'avant dernière règle.

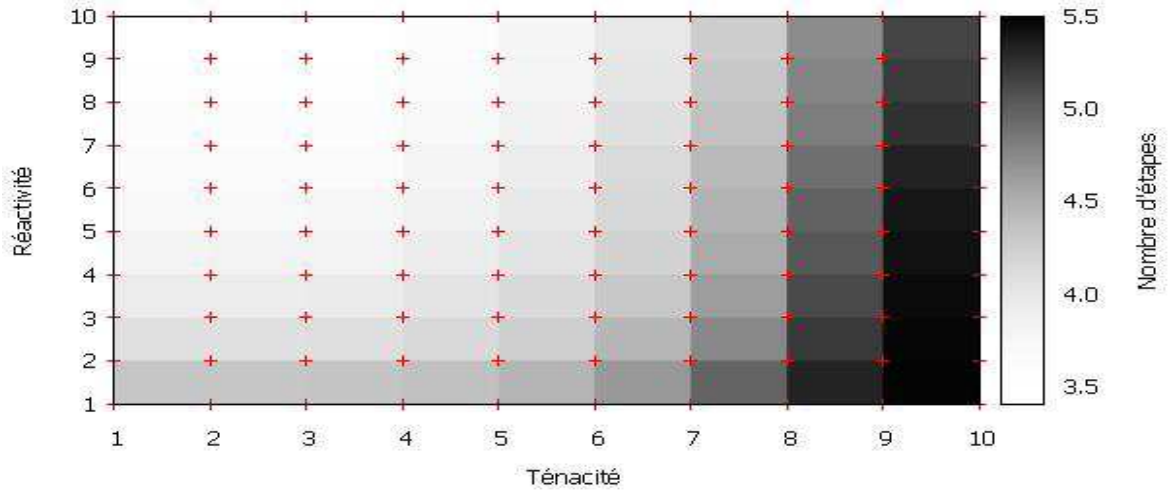


Figure 4.30. *Le logarithme de la moyenne des nombres de pas nécessaires pour la convergence, en fonction de la ténacité et la réactivité des deux acteurs. Le nombre de pas varie entre environ 3 000 et 300 000.*

La figure 4.30 indique le logarithme du nombre de pas pour avoir une échelle de valeur plus précise. Son examen montre que plus la ténacité augmente et la réactivité diminue, plus le nombre de pas nécessaire pour la convergence augmente. Reste à noter qu'aucune simulation n'a convergé pour une ténacité de 10 et une réactivité de 1 en moins de 300 000 pas, ce qui explique la valeur nulle pour la satisfaction dans la figure 4.31. En deçà de 5, la ténacité ne semble pas avoir une influence significative.

Seule la satisfaction de l'acteur A est présentée dans la figure 4.31 car les satisfactions des deux acteurs sont semblables quelque soit la valeur de la ténacité et de la réactivité : cette satisfaction, peu affectée par la réactivité sauf pour une ténacité moyenne entre 5 et 8, augmente avec la ténacité et diminue avec la réactivité.

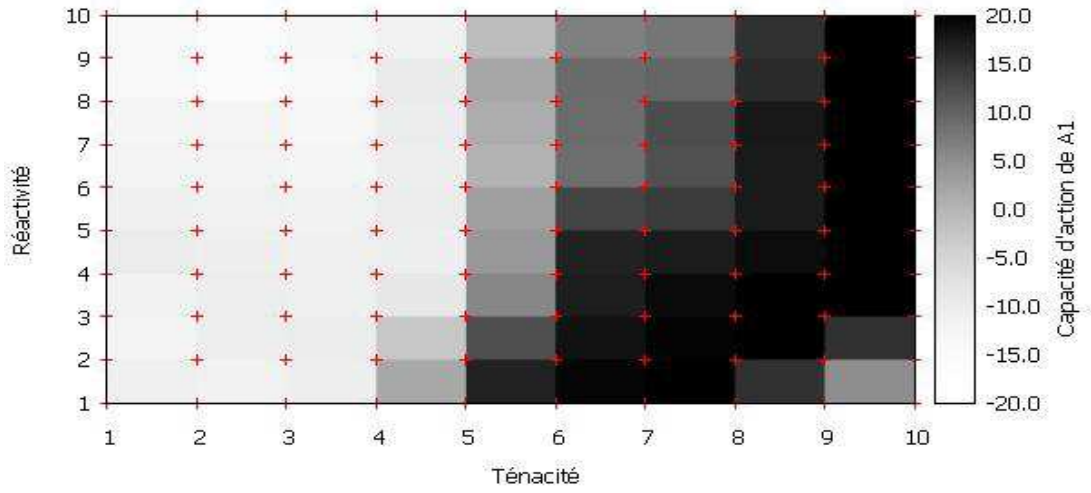


Figure 4.31. *La moyenne de la satisfaction de chaque acteur, en fonction de la réactivité des deux acteurs.*

Par conséquent, on peut déduire que le couple (ténacité, réactivité) de (6, 1) ou (7, 2) pour les deux acteurs conduit à une meilleure convergence pour ce jeu : une satisfaction sociale maximale pour les deux acteurs, et une simulation de 41633 pour le couple (6, 1) et 41789 étapes pour le couple (7, 2).

Le modèle free-rider

Les figures 4.32 et 4.33 montrent les résultats d'une analyse de sensibilité du modèle free rider, comprenant 100 expériences où la ténacité et la réactivité de l'acteur Act1 varient entre 1 et

10, tandis que la ténacité et la réactivité des autres acteurs sont fixées à 5. Chaque expérience d'analyse de sensibilité a été réalisée avec 100 simulations. Les paramètres psycho-cognitifs sont initialisés de façon standard : discernement = 1, et une répartition de 10% pour la dernière règle et 90% pour l'avant dernière règle.

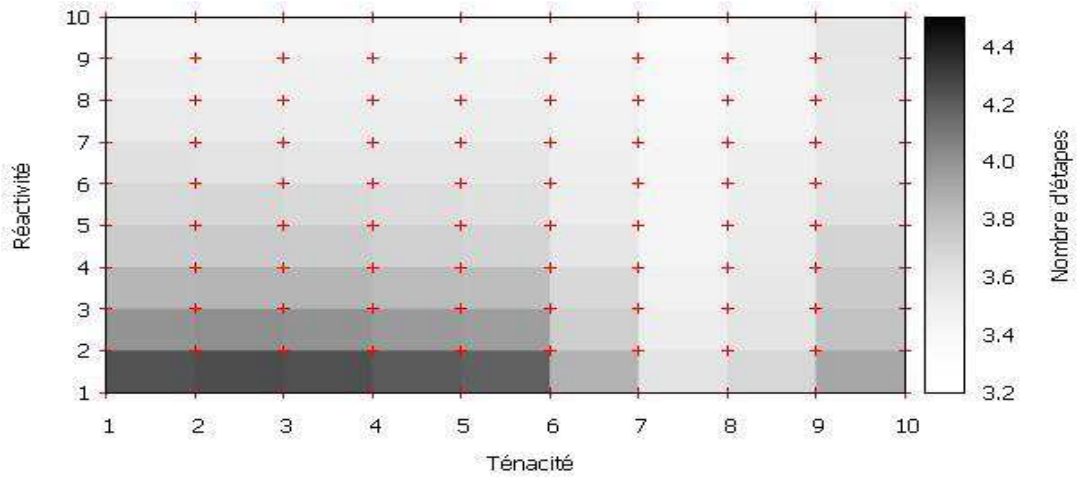


Figure 4.32. *Le logarithme de la moyenne des nombres de pas nécessaires pour la convergence, en fonction de la ténacité et de la réactivité de l'acteur Act1. Le nombre de pas varie entre environ 1 700 et 26 000.*

Le nombre de pas est sensiblement affecté par la réactivité de l'acteur Act1. Par contre la ténacité n'a peu effet sur le nombre de pas (cf. figure 4.32).

En ce qui concerne la satisfaction de A1 (cf. figure 4.33), plus la ténacité de A1 augmente, plus sa satisfaction augmente. Par contre, sa réactivité n'a aucun effet sur sa satisfaction.

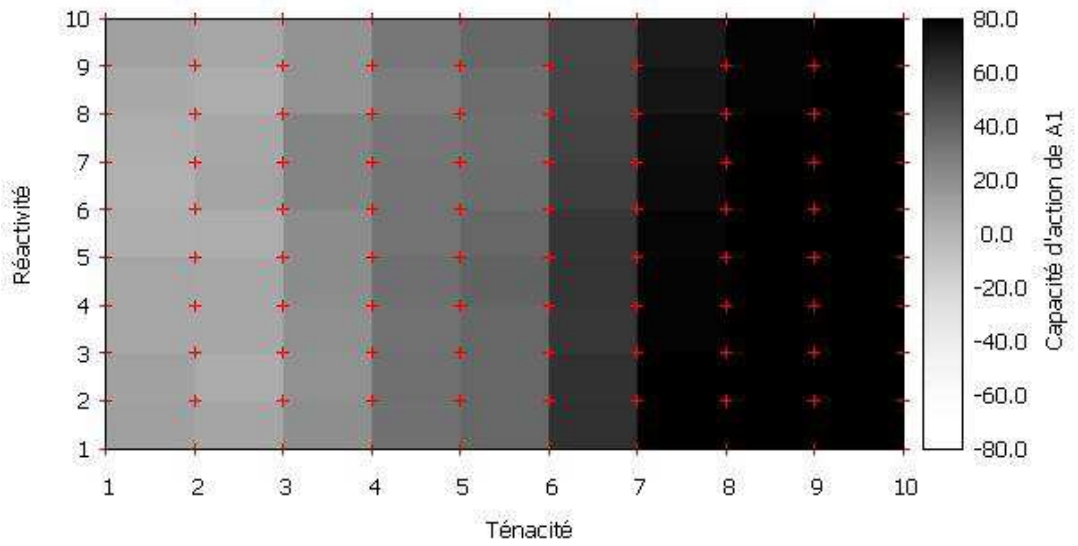


Figure 4.33. *La moyenne de la satisfaction de chaque acteur (Act1 Jaune, Act2 Vert, Act3 Bleu, Act4 Rouge), en fonction de la ténacité de l'acteur Act1.*

Par conséquent, on peut déduire que le couple (ténacité, réactivité) de (10, 8) pour l'acteur Act1 conduit à une meilleure convergence pour ce jeu : une satisfaction sociale maximale de A1, et une simulation de 1707 étapes.

Discussion

En ce qui concerne la satisfaction des acteurs, dans les deux cas considérés ci-dessus, la ténacité influence fortement cette satisfaction (cf. figure 4.31 et 4.33) quelque soit la valeur de la

réactivité, tandis que la réactivité les influence pour des valeurs moyennes de la ténacité dans le dilemme du prisonnier 4/6 (cf. figure 4.31) et n'a aucune influence dans le modèle free-rider (cf. figure 4.33). Dans d'autres cas où la structure du jeu impose la coopération entre les acteurs, par exemple le dilemme du prisonnier 1/9, la ténacité et la réactivité n'ont aucune influence, les acteurs doivent coopérer afin d'augmenter leur satisfaction.

En ce qui concerne la durée des simulations, dans tous les cas et plus précisément dans les deux cas considérés ci-dessus, la réactivité et la ténacité influence légèrement cette durée (cf. figures 4.30 et 4.32) sauf pour une ténacité proche de sa valeur maximale ou une réactivité proche de sa valeur minimale. En effet, pour une ténacité maximale ou une réactivité minimale, l'ambition d'un acteur diminue de façon quasi négligeable, ce qui augmente largement la durée de simulation car un état de convergence n'est atteint que si la satisfaction a dépassé l'ambition de tous les acteurs.

Dans tous les cas, le couple (ténacité, réactivité) choisi par le modélisateur doit être un compromis entre les deux effets de la variation de ces deux paramètres : une simulation plus coûteuse en temps de calcul, et une coopération plus élevée. Pour ce faire, nous pouvons considérer la même fonction que précédemment (cf. éq. 4.20), qui mesure, en proportion, l'écart entre la satisfaction d'un acteur et la durée des simulations par rapport à leurs valeurs maximales et minimales :

$$InfluenceTénRéa = \left(\frac{satisfaction - satisfactionMin}{satisfactionMax - satisfactionMin}\right) - \left(\frac{nbEtapes - nbEtapesMin}{nbEtapesMax - nbEtapesMin}\right) \quad \text{(éq. 4.20)}$$

où satisfactionMax, satisfactionMin, NbEtapesMax, et NbEtapesMin sont les valeurs maximales et minimales obtenues à la fin des simulations. Un « bon » couple (ténacité, réactivité) est celui pour lequel cette fonction dépasse un certain seuil (par exemple fixé à 0,8 dans le cas du dilemme du prisonnier, cf. figure 4.34), tandis que le meilleur couple est celui pour lequel cette fonction est maximale (pour un couple (ténacité, réactivité) de (7, 2) dans le cas du dilemme du prisonnier, cf. figure 4.34).

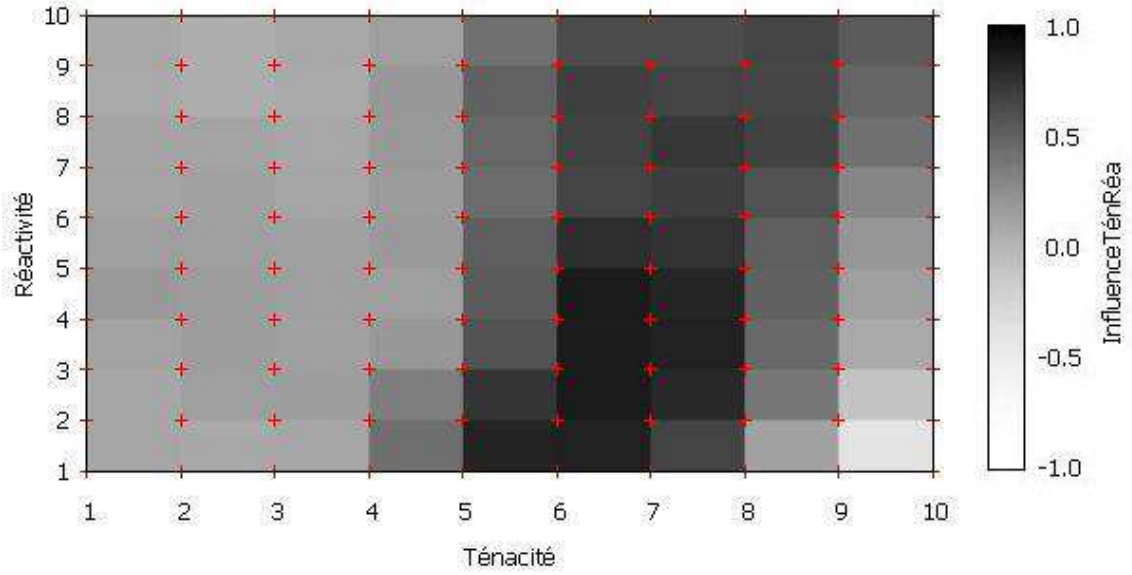


Figure 4.34. *La courbe de l'InfluenceTénRéa en fonction de la ténacité et la réactivité des deux acteurs dans le dilemme du prisonnier.*

4.5.4 – Analyse du discernement

Le discernement d'un acteur détermine sa capacité à distinguer entre les différentes situations selon leur plus ou moins grande proximité (cf. section 4.2.1). Ce paramètre est lié à la structure du modèle. En effet, l'importance de la valeur de ce paramètre dépend de la nature des fonctions

d'effet de chaque acteur : pour un acteur ayant toutes ses fonctions d'effet linéaires, toutes les situations sont semblables puisque la même action a toujours le même effet. De ce point de vue, toutes les situations sont similaires. Par contre, un acteur ayant une fonction d'effet en forme parabolique aura besoin de distinguer entre les situations pour lesquelles une même action augmente ou diminue l'effet.

Pour étudier l'influence du discernement sur le nombre de pas nécessaires pour la convergence et le niveau de satisfaction acquise par les acteurs, nous analysons les résultats de l'algorithme de simulation sur deux modèles d'organisation : le dilemme du prisonnier, et un modèle à deux acteurs et six relations.

Le dilemme du prisonnier 4/6

Le dilemme du prisonnier traité dans cette analyse est celui avec une répartition 4 / 6. Les figures 4.35 et 4.36 montrent les résultats d'une analyse de sensibilité, comprenant 5 expériences où le discernement est le même pour les deux acteurs et varie entre 1 et 5. Chaque expérience d'analyse de sensibilité a été réalisée avec 100 simulations. Les paramètres psycho-cognitifs sont initialisés de façon standard : ténacité = 5, réactivité = 5, et une répartition de 40% pour la dernière règle et 60% pour l'avant dernière règle.

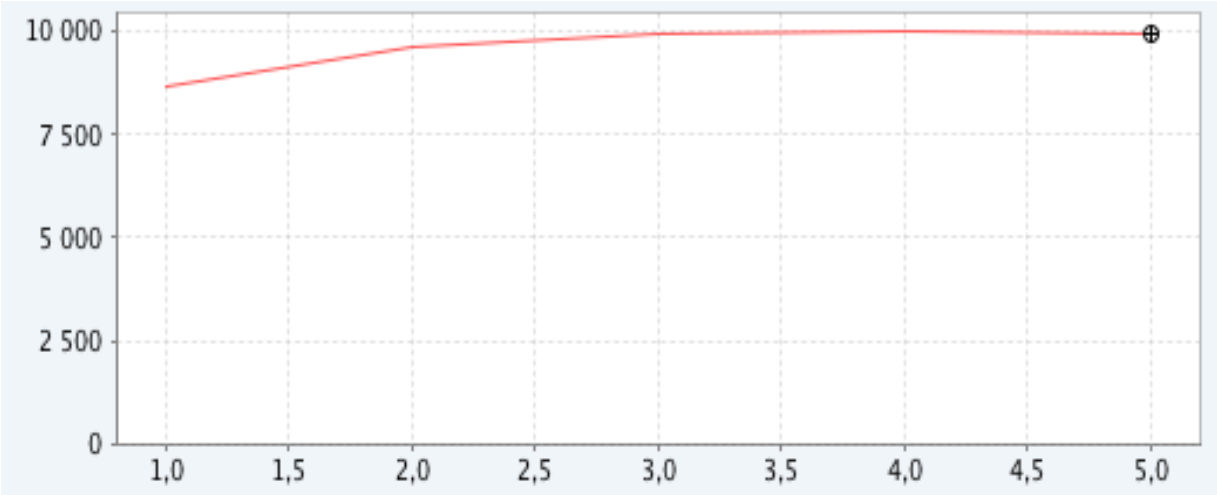


Figure 4.35. *La moyenne du nombre de pas nécessaires pour la convergence, en fonction du discernement des deux acteurs dans le dilemme du prisonnier 4/6.*

Le nombre de pas nécessaires pour la convergence (cf. figure 4.35) augmente légèrement d'environ 8 600 à 10 000 quand le discernement des deux acteurs augmente de 1 à 5.

Les satisfactions des deux acteurs (cf. figure 4.36) sont semblables quelque soit leur discernement. Elles diminuent d'environ -3 à -20 quand le discernement augmente de 1 à 3, et restent constante à -20 pour un discernement de 4 ou 5.

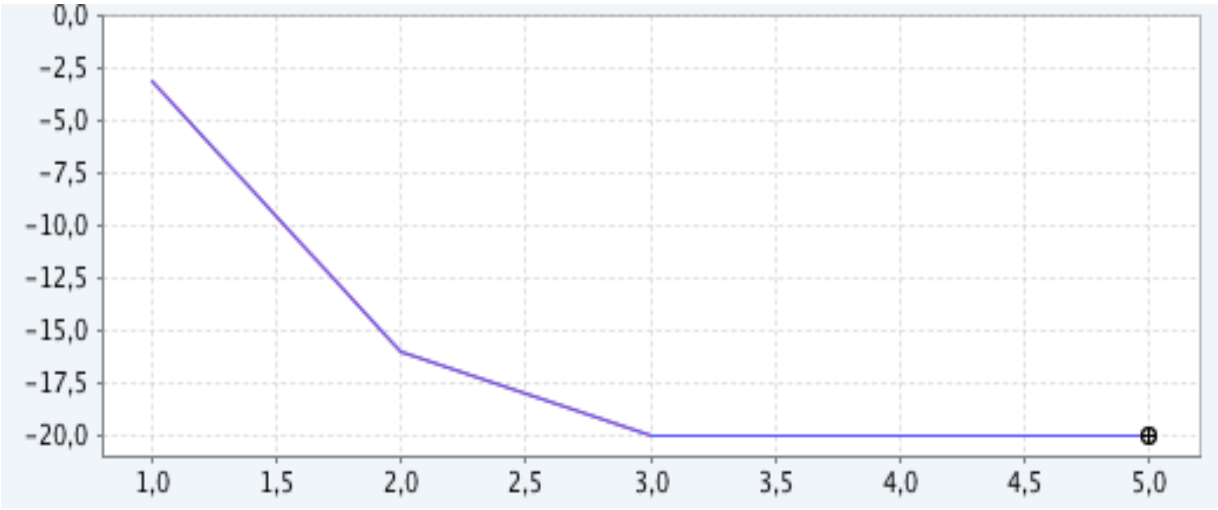


Figure 4.36. *La moyenne de la satisfaction de chaque acteur, en fonction du discernement des deux acteurs dans le dilemme du prisonnier 4/6.*

Ces résultats peuvent être expliqués : pour un discernement plus grand que 1, les acteurs font une distinction entre des situations qui, en fait, sont identiques en variabilité puisque les fonctions d'effets sont linéaires. Ils renouvellent donc les règles de leur base et il semble qu'ils n'arrivent pas à évaluer correctement la qualité de ces règles. Plus le discernement augmente, plus les acteurs semblent confus, et plus leurs satisfactions diminuent.

Nous avons aussi considéré le cas du dilemme du prisonnier pour une répartition d'enjeux 3,5 sur la relation que l'acteur contrôle et 6,5 sur la relation contrôlée par l'autre acteur. Les satisfaction des acteurs sont constantes à 30 (30 étant la satisfaction de chaque acteur dans la configuration correspondant au maximum de la satisfaction globale) pour un discernement des deux acteurs entre 1 et 4, et diminuent très légèrement à 29,2 pour un discernement de 5. Le nombre de pas nécessaires pour la convergence augmente de façon exponentielle de 7 000 à 272 000 quand le discernement augmente de 1 à 5. En deçà de 3 d'enjeux sur la relation que l'acteur contrôle, le discernement n'a aucun effet sur les satisfactions des acteurs : elles sont constantes et égales aux valeurs maximales quelque soit le discernement des deux acteurs : 40 pour la répartition 3/7, 60 pour la répartition 2/8, 80 pour la répartition 1/9 et 100 pour la répartition 0/10. Le nombre de pas nécessaires pour la convergence varie de la même façon dans les quatre cas : il augmente de 6 200 à 79 000 pour la répartition 3/7, de 4 000 à 15 000 pour la répartition 2/8, de 1 000 à 7200 pour la répartition 1/9, et de 300 à 800 pour la répartition 0/10.

Nous avons aussi considéré le cas du dilemme du prisonnier 4/6 où les valeurs optimales de la ténacité et de la réactivité ont été utilisées, c'est-à-dire pour une ténacité de 7 et une réactivité de 2 pour les deux acteurs. Le nombre de pas nécessaires pour la convergence (cf. figure 4.37) varie de façon parabolique : il augmente d'environ 75 000 à 200 000 quand le discernement des deux acteurs augmente de 1 à 3, et diminue jusqu'à 120 000 quand le discernement augmente jusqu'à 5. Pour un discernement élevé (4 ou 5), les acteurs n'explorent plus suffisamment, ce qui explique la diminution du nombre de pas. Les satisfactions des deux acteurs (cf. figure 4.38) sont semblables quelque soit leur discernement. Elles diminuent de façon sigmoïde de 20 à -20 quand le discernement des deux acteurs augmente de 1 à 5. Remarquons qu'une valeur inadéquate pour le discernement rend inefficace les valeurs optimales de la ténacité et de la réactivité choisies pour cette analyse.

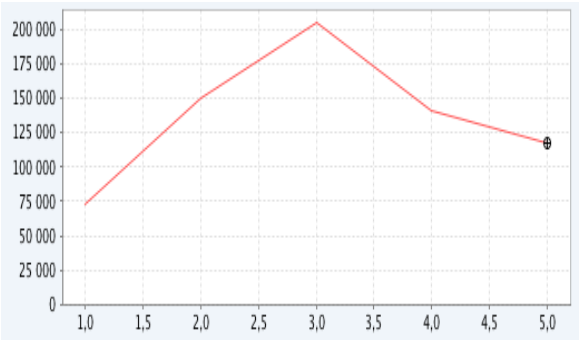


Figure 4.37. *La moyenne du nombre de pas nécessaires pour la convergence, en fonction du discernement des deux acteurs.*

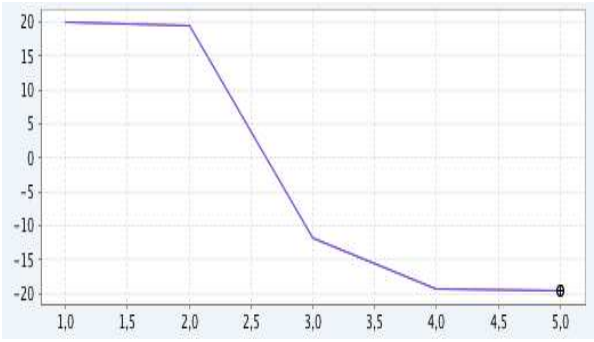


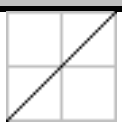
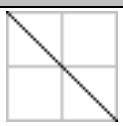
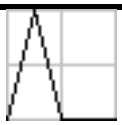
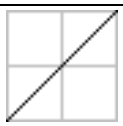
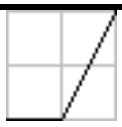
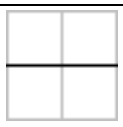
Figure 4.38. *La moyenne de la satisfaction de chaque acteur, en fonction du discernement des deux acteurs.*

Un modèle à deux acteurs et six relations

Le modèle d'organisation traité dans cette section comporte deux acteurs et six relations (cf. tableau 4.12) : chaque acteur contrôle trois relations, et dépend aussi de deux autres relations contrôlées par l'autre acteur. Il met 90% de ses enjeux sur les relations qu'il contrôle et 10% sur les autres relations (cf. tableau 4.12). Les fonctions d'effets (cf. tableau 4.13) sont définies de façon à avoir un effet sur la satisfaction de l'acteur non-monotone en fonction des états des relations dont il dépend, ce qui rend utile de distinguer entre plusieurs situations pour maximiser sa satisfaction. Ce modèle parfaitement symétrique a été construit de façon très simple pour clarifier l'effet du discernement des acteurs.

	A1	A2
R11	**3**	0,5
R12	**3**	0,5
R13	**3**	0
R21	0,5	**3**
R22	0,5	**3**
R23	0	**3**

Tableau 4.12. *Les enjeux des acteurs sur les relations (les contours épais indiquent le contrôleur de la relation).*

	A1	A2
R11		
R12		
R13		

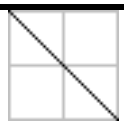
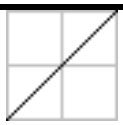
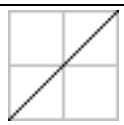
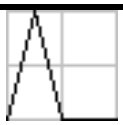
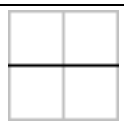
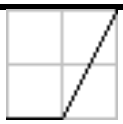
R21		
R22		
R23		

Tableau 4.13. *Les fonctions d'effet des relations sur les acteurs.*

Les figures 4.39 et 4.40 montrent les résultats d'une analyse de sensibilité de ce modèle, comprenant 5 expériences où le discernement est le même pour les deux acteurs et varie entre 1 et 5. Chaque expérience d'analyse de sensibilité a été réalisée avec 100 simulations. Les paramètres psycho-cognitifs sont initialisés de façon standard : ténacité = 5, réactivité = 5, et une répartition de 90% pour la dernière règle et 10% pour l'avant dernière règle.

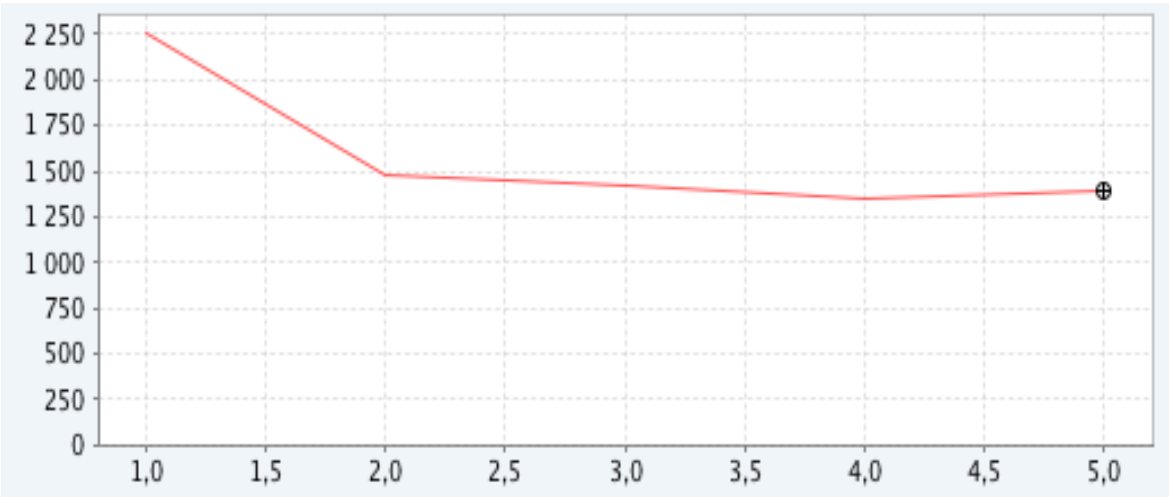


Figure 4.39. *La moyenne des nombres de pas nécessaires pour la convergence, en fonction du discernement des deux acteurs dans le modèle à deux acteurs et six relations.*

Le nombre de pas nécessaires pour la convergence diminue d'environ 2 250 à 1 500 quand le discernement augmente de 1 à 2, et diminue très légèrement jusqu'à 1 400 quand le discernement augmente jusqu'à 5 (cf. figure 39).

Les satisfactions des deux acteurs (cf. figure 4.40) sont semblables quelque soit leur discernement. Elles augmente d'environ 64 à 75 quand le discernement augmente de 1 à 2, et reste quasiment constante à -75,3 quand le discernement augmente jusqu'à 5.

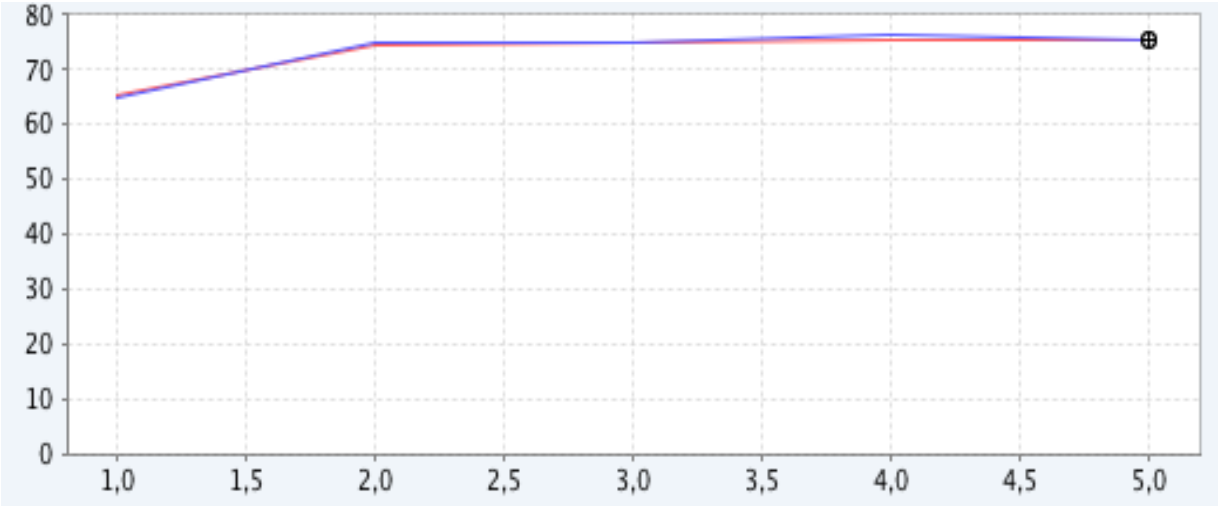


Figure 4.40. *La moyenne de la satisfaction de chaque acteur en fonction du discernement des deux acteurs dans le modèle à deux acteurs et six relations.*

Le modèle présenté dans ce paragraphe exige de l'acteur une bonne capacité à distinguer entre les situations afin d'améliorer sa satisfaction. Mais remarquons qu'au delà de 3 pour le discernement, les résultats de simulation sont semblables.

Discussion

Le discernement d'un acteur détermine sa capacité à discriminer les situations. Plus précisément, il détermine le rayon d'applicabilité d'une règle sous forme d'un seuil : une règle n'est plus applicable si la distance euclidienne entre la situation courante de l'acteur et celle de la règle est supérieure au seuil.

$$seuil = \text{distance (situationMax, situationMin)} / discernement$$

L'augmentation du discernement d'un acteur produit une meilleure capacité à discriminer les situations, qui à son tour produit une augmentation de la satisfaction des acteurs dans les cas où plusieurs situations différentes existent (c'est-à-dire la même action produit une augmentation de la satisfaction dans un certain état et une diminution dans un autre état (cf. figure 4.40)), et une diminution de la satisfaction dans les cas contraires où toutes les situations sont similaires (c'est-à-dire une action a toujours le même effet dans tous les états (cf. figure 4.36 et 4.38)). Cependant, dans certaines organisations où la coopération est imposée par la structure, par exemple le dilemme du prisonnier avec une répartition 1 / 9, le discernement n'a aucun effet sur les résultats de simulation.

Inversement à son effet sur les satisfactions, l'augmentation du discernement d'un acteur produit une augmentation de la durée des simulations dans les cas où une capacité de discrimination élevée est inutile (c'est-à-dire n'augmente pas la satisfaction, cf. figure 4.35), et une diminution de cette durée dans les cas contraires (cf. figure 4.39).

Dans tous les cas, une valeur du paramètre discernement doit être choisie par le modélisateur en fonction de la présence de situations différentes, plus précisément en fonction de la nature des fonctions d'effet de chaque acteur. Par exemple, un acteur ayant une fonction d'effet de forme parabolique (cf. tableau 4.13) aura besoin de distinguer entre situations différentes, car une même action augmente sa satisfaction dans certains états et la diminue dans d'autres.

4.5.5 – Analyse de la répartition du renforcement

Lorsqu'un acteur exécute l'action d'une règle, il perçoit immédiatement à l'étape suivante l'impact direct de cette action sur sa satisfaction, mais il lui faut attendre l'étape suivante pour que sa satisfaction enregistre la façon dont les acteurs dont il dépend ont réagi à cette action. La

répartition du renforcement permet à l'acteur de répartir la mise à jour de la qualité de l'action effectuée à un instant t sur la variation de sa satisfaction aux deux étapes consécutives d'instants t+1 et t+2. Ce paramètre dépend fortement de la structure du jeu, plus précisément de la distribution des enjeux de l'acteur sur les relations dont il dépend. En effet, un acteur, qui contrôle toutes les relations dont il dépend n'a pas besoin d'attendre à l'étape t+2 pour connaître l'effet de ses actions sur sa satisfaction. Inversement, un acteur qui ne contrôle qu'une très faible part des relations dont il dépend est obligé d'évaluer ses actions à l'étape t+2.

Le paramètre « répartition du renforcement » détermine, lors de la mise à jour à l'étape t de la qualité des règles en fonction de la variation de la satisfaction de l'acteur, le pourcentage de cette variation qui est attribué à la règle appliquée à l'étape t-1, le pourcentage complémentaire étant attribué à la règle appliquée à l'étape t-2.

Pour étudier l'influence de ce paramètre sur le nombre de pas nécessaires pour la convergence et le niveau de satisfaction obtenue par les acteurs, nous analysons les résultats de l'algorithme de simulation sur deux modèles d'organisation : le dilemme du prisonnier, et le modèle free rider.

Le dilemme du prisonnier 4/6

Le dilemme du prisonnier traité dans cette analyse est celui avec une répartition 4 / 6. Les figures 4.41 et 4.42 montrent les résultats d'une analyse de sensibilité, comprenant 11 expériences où la répartition du renforcement est la même pour les deux acteurs et varie entre 0% et 100%. Chaque expérience d'analyse de sensibilité a été réalisée avec 100 simulations. Les paramètres psycho-cognitifs sont initialisés de façon standard: discernement = 1, et ténacité = 5, réactivité = 5.

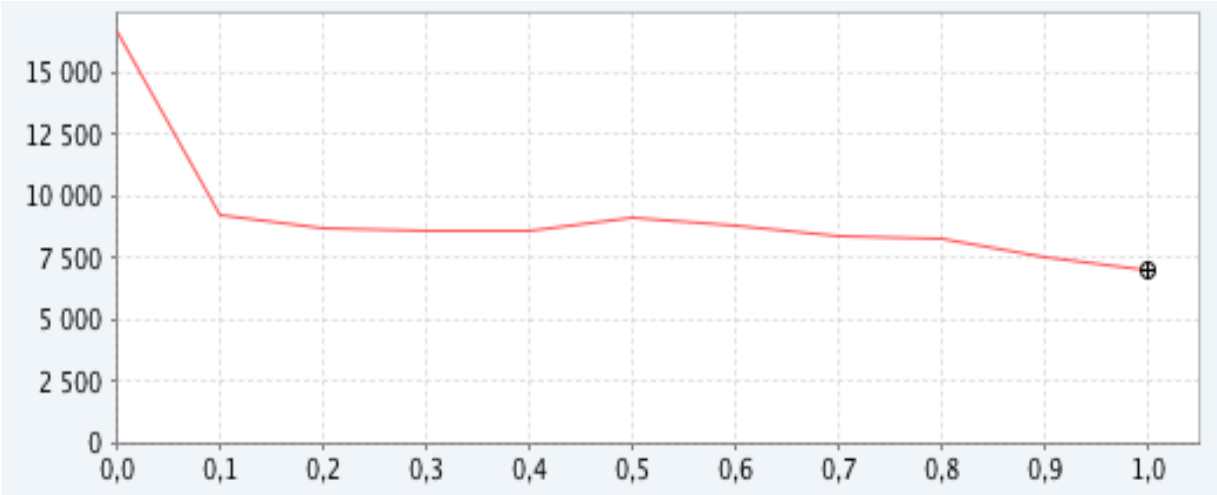


Figure 4.41. *La moyenne des nombres de pas nécessaires pour la convergence, en fonction de la répartition du renforcement des deux acteurs dans le dilemme du prisonnier 4/6.*

La figure 4.41 montre que le nombre de pas nécessaires pour la convergence diminue brusquement d'environ 16 500 à 9 200 quand la répartition du renforcement augmente de 0 à 10%, et puis diminue légèrement jusqu'à 7 000 quand la répartition du renforcement augmente jusqu'à 100% sur la dernière règle.

Les satisfactions des deux acteurs (cf. figure 4.42) sont semblables quelque soit la valeur de la répartition du renforcement. Elles diminuent d'environ 20 à -7 quand la répartition du renforcement augmente de 0% jusqu'à 50%, et augmentent jusqu'à environ 12 quand la répartition du renforcement augmente jusqu'à 100% sur la dernière règle.

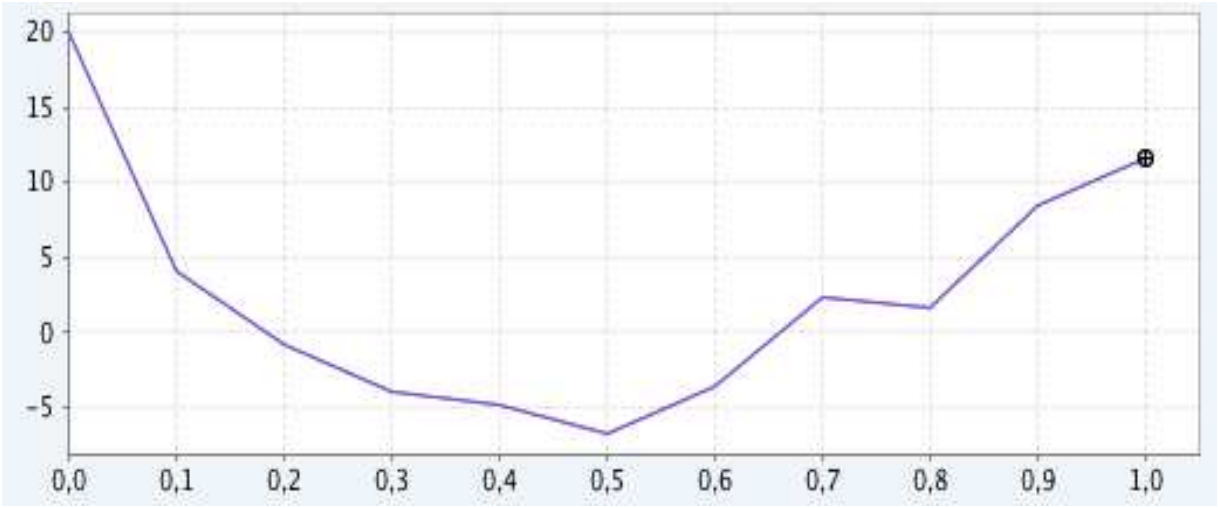


Figure 4.42. *La moyenne de la satisfaction de chaque acteur, en fonction de la répartition du renforcement des deux acteurs dans le dilemme du prisonnier 4/6.*

Dans le dilemme du prisonnier à deux acteurs, la coopération de chacun des acteurs est indispensable à l'obtention d'une bonne satisfaction pour chacun d'eux. Cette coopération conjointe est le fruit du hasard : il se trouve que tout deux coopèrent, s'en trouvent bien, et donc poursuivent dans le même sens. Pour une répartition de 0/10 d'enjeux, il suffit que les deux acteurs appliquent une action coopérative (donc positive) afin que les deux acteurs détectent le gain résultant de la coopération en percevant l'augmentation de leur satisfaction. Pour une répartition de 1/9, deux actions positives ne conduisent pas toujours à une augmentation de la satisfaction de chacun des deux acteurs. Par exemple, si A1 choisit une action positive a1 et A2 choisit une action positive a2, il faut que les deux actions respectent les conditions ($a1 < a2 * \varepsilon$ et $a2 < a1 * \varepsilon$, avec $\varepsilon = 9$) afin que la satisfaction des deux acteurs augmente. De ce fait, plus le nombre d'enjeux posés par l'acteur sur la relation qu'il contrôle augmente, plus ε diminue, et par conséquent plus la probabilité de choisir simultanément deux actions qui vérifient ces conditions diminue, ce qui rend la coopération difficile à trouver dans un dilemme du prisonnier 4/6.

En ce qui concerne la figure 4.42, pour une répartition de 0% ou 100% sur la dernière règle, il suffit que les deux acteurs choisissent au même instant deux actions remplissant les deux conditions ($a1 < a2 * \varepsilon$ et $a2 < a1 * \varepsilon$, avec $\varepsilon = 1,5$) pour que la satisfaction des deux acteurs augmentent, et ainsi les deux actions seront considérées comme « bonne ». Par contre, pour une répartition de 0,5, l'action d'une règle entre pour bien peu dans le différentiel de satisfaction qui sera utilisé pour renforcer cette règle. Par conséquent, les acteurs sont obligés de choisir en même temps deux actions positives « consécutives » remplissant les deux conditions ($a1 < a2 * \varepsilon$ et $a2 < a1 * \varepsilon$, avec $\varepsilon = 1,5$) afin de trouver la coopération profitable, ce qui semble être très difficile. Ainsi on peut conclure pour ce jeu que plus l'acteur mélange l'effet immédiat et l'effet différé de ses actions sur sa satisfaction, plus il évalue mal ses règles, et plus sa satisfaction diminue. Remarquons que la meilleure valeur de satisfaction est obtenue pour une récompense nulle sur la dernière règle appliquée, ce qui est cohérent avec le fait que l'acteur dépend davantage de l'action de l'autre acteur que de la sienne propre.

Nous avons aussi considéré le cas du dilemme du prisonnier pour une répartition d'enjeux 3,5 sur la relation que l'acteur contrôle, et 6,5 sur la relation contrôlée par l'autre acteur. Les résultats sont assez comparables au cas précédent comme illustré dans les figures 4.43 et 4.44:

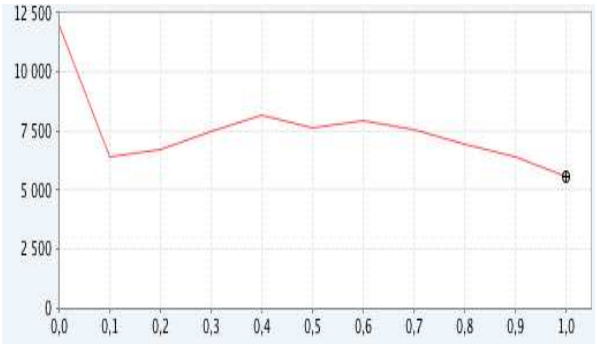


Figure 4.43. *La moyenne des nombres de pas nécessaires pour la convergence, en fonction de la répartition du renforcement de A1 dans le dilemme du prisonnier 4/6.*

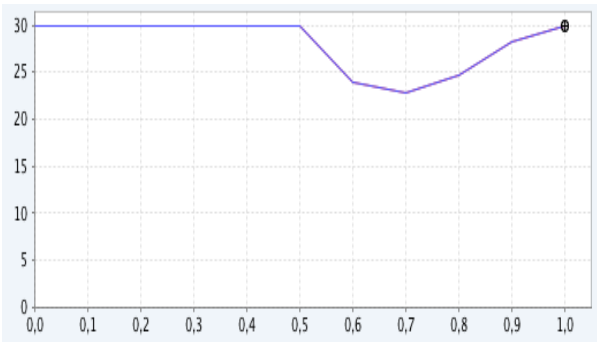


Figure 4.44. *La moyenne de la satisfaction de chaque acteur, en fonction de la répartition du renforcement de A1 dans le dilemme du prisonnier 4/6.*

En deçà de 3 points d'enjeux sur la relation que l'acteur contrôle, la répartition n'a aucun effet sur les satisfactions des acteurs : elles sont constantes et maximales quelque soit la répartition du renforcement des deux acteurs. Dans les quatre cas de répartition d'enjeux, le nombre de pas nécessaires pour la convergence varie de la même façon que le cas précédent (cf. figure 4.43).

Nous avons aussi considéré le cas du dilemme du prisonnier 4/6 où les valeurs optimales des autres paramètres psycho-cognitifs ont été utilisées, c'est-à-dire une ténacité de 7, une réactivité de 2, et un discernement de 1 pour les deux acteurs. La figure 4.45 montre que le nombre de pas nécessaires pour la convergence diminue de façon sigmoïdale de 500 000 à environ 50 000 pas quand la répartition du renforcement augmente de 0% à 100%. Les satisfactions des deux acteurs (cf. figure 4.46) sont presque semblables quelque soit la valeur de la répartition du renforcement, et sont quasiment constantes égales à 20. Pour une répartition du renforcement nulle sur la dernière règle, aucune simulation n'a convergé, et la moyenne des satisfactions est 0 par défaut. Avec un couple (ténacité, réactivité) optimal, la répartition du renforcement n'a aucune influence sur les satisfactions des acteurs, mais elle conduit à des simulations très coûteuses quand l'acteur considère peu l'effet immédiat de ses actions sur sa satisfaction (répartition proche de 0).

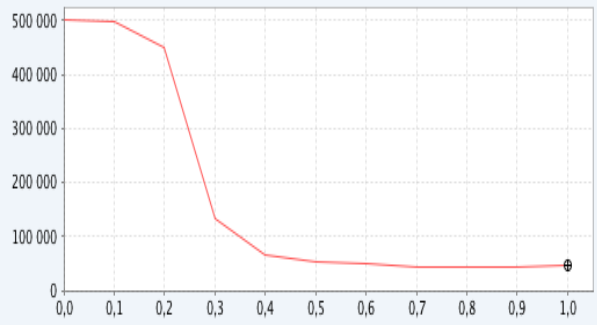


Figure 4.45. *La moyenne du nombre de pas nécessaires pour la convergence, en fonction de la répartition du renforcement des deux acteurs dans le dilemme du prisonnier 4/6.*

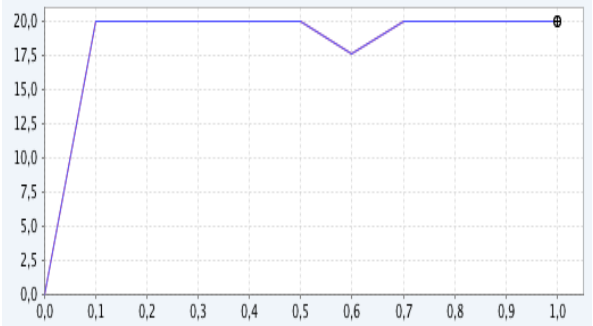


Figure 4.46. *La moyenne de la satisfaction de chaque acteur, en fonction de la répartition du renforcement des deux acteurs dans le dilemme du prisonnier 4/6.*

Le modèle free-rider

Les figures 4.47 et 4.48 montrent les résultats d'une analyse de sensibilité du modèle free rider, comprenant 11 expériences, où la répartition du renforcement de l'acteur Act1 varie entre 0% et 100 sur la dernière règle, tandis que la répartition du renforcement des trois autres est fixée à 90% pour la dernière règle et 10% pour l'avant dernière règle. Chaque expérience d'analyse de sensibilité a été réalisée avec 100 simulations. Les paramètres psycho-cognitifs sont initialisés de façon standard : discernement = 1, ténacité = réactivité = 5.

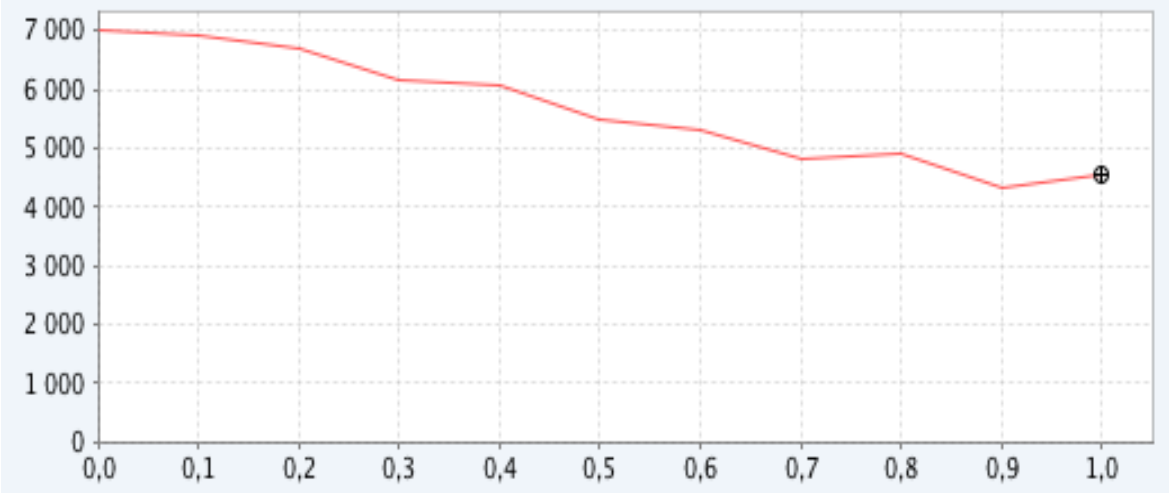


Figure 4.47. *La moyenne des nombres de pas nécessaires pour la convergence, en fonction de la répartition du renforcement de l'acteur Act1 dans le modèle free-rider.*

Le nombre de pas nécessaires pour la convergence diminue de façon linéaire de 7 000 à 4 500 avec l'augmentation de la répartition du renforcement de l'acteur Act1 (cf. figure 4.47).

La satisfaction de l'acteur A1 augmente aussi de façon quasiment linéaire d'environ 22 à 35 avec l'augmentation de sa répartition du renforcement, tandis que les satisfaction des trois autres acteur diminue très légèrement (cf. figure 4.48).

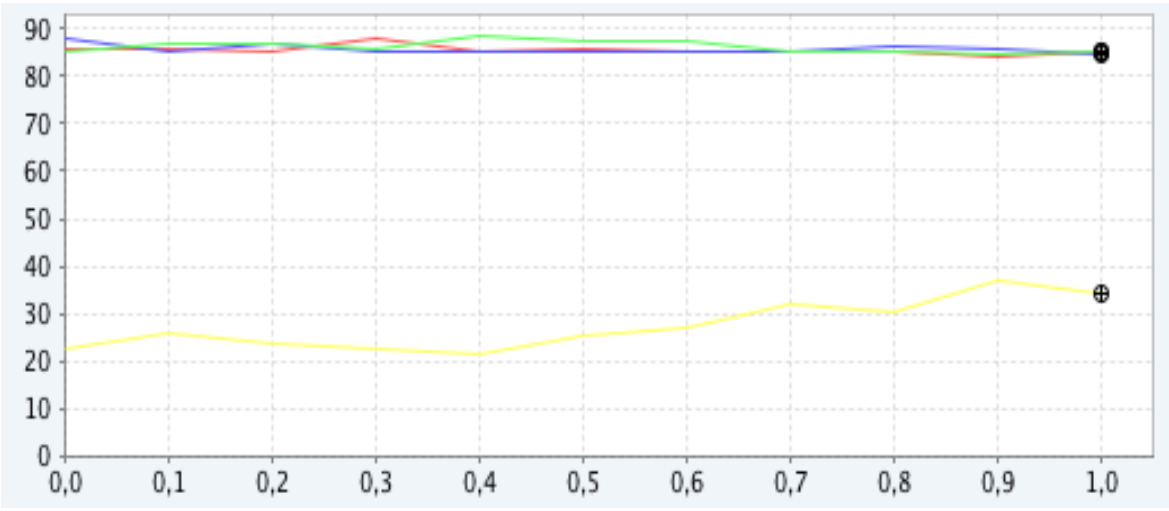


Figure 4.48. *La moyenne de la satisfaction de chaque acteur (Act1 Jaune, Act2 Vert, Act3 Bleu, Act4 Rouge), en fonction de la répartition du renforcement de l'acteur Act1 dans le modèle free-rider.*

Donnons une interprétation à ces résultats : l'acteur Act1 cherche à tout prix à convaincre les trois acteurs de coopérer complètement avec lui. Plus il considère l'effet direct de ses actions sur sa satisfaction, plus il exige la coopération des autres acteurs au même instant, et plus il améliore sa satisfaction.

Nous avons aussi considéré le cas où tous les acteurs ont la même répartition du renforcement qui varie entre 0% et 100%. Les résultats sont comparables au cas précédent où seul la répartition du renforcement de A1 varie : les satisfactions des trois acteurs (Act2, Act3, et Act4) diminue légèrement, tandis que la satisfaction de l'acteur Act1 augmente avec l'augmentation du répartition du renforcement des acteurs (cf. figure 4.50). Le nombre de pas nécessaires pour la convergence diminue d'environ 8 300 à 4 000 avec l'augmentation du répartition du renforcement des acteurs (cf. figure 4.49).

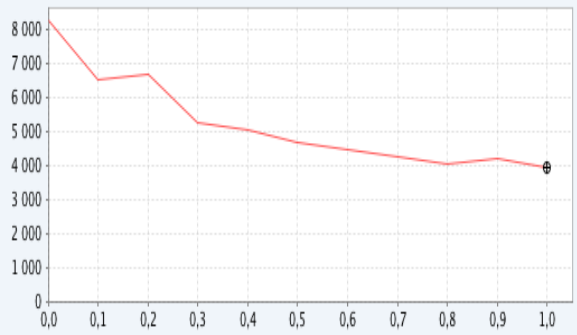


Figure 4.49. *La moyenne des nombres de pas nécessaires pour la convergence, en fonction de la répartition du renforcement de tous les acteurs dans le modèle free-rider.*

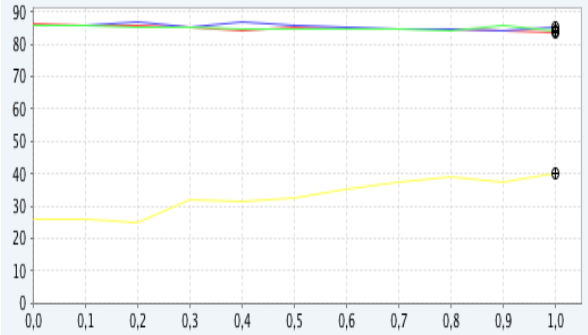


Figure 4.50. *La moyenne de la satisfaction de chaque acteur (Act1 Jaune, Act2 Vert, Act3 Bleu, Act4 Rouge), en fonction de la répartition du renforcement de tous les acteurs dans le modèle free-rider.*

Discussion

La répartition du renforcement permet à un acteur d'attacher plus ou moins d'importance à la réaction des autres acteurs. Elle permet de favoriser la détection de la coopération si cette dernière est peu profitable. Elle intervient dans la mise à jour des forces des règles de la façon suivante :

$$RA_{t\text{-}1}.force_t = (1 - TX_t) \times RA_{t\text{-}1}.force_{t\text{-}1} + TX_t \times PRR \times \Delta satisfaction_t$$

$$RA_{t\text{-}2}.force_t = RA_{t\text{-}2}.force_{t\text{-}1} + TX_t \times (1 - PRR) \times \Delta satisfaction_t$$

En ce qui concerne la satisfaction des acteurs dans les cas où la coopération est très profitable aux acteurs, par exemple le dilemme du prisonnier avec une répartition d'enjeux 1/9, la répartition du renforcement n'a aucun effet sur les satisfactions des acteurs. Par contre, dans le cas du dilemme du prisonnier 4/6 où la coopération est peu profitable, une valeur intermédiaire de ce paramètre embrouille l'évaluation des règles et diminue les satisfactions des acteurs, tandis qu'évaluer les règles uniquement à l'étape t+1 ou à l'étape t+2, cause une augmentation des satisfactions. En outre, dans le modèle free-rider, plus l'acteur A1 considère l'effet direct de ses actions sur sa satisfaction, plus il exige la coopération des trois autres au même instant, plus la probabilité d'avoir un acteur parmi les trois qui joue la trahison diminue, et plus sa satisfaction augmente.

En ce qui concerne la durée des simulations, l'effet de ce paramètre est négligeable, sauf pour des valeurs proches de la valeur minimale (cf. figure 4.45). Pour ces valeurs, l'acteur donne plus d'importance à l'effet indirect de ses actions, c'est-à-dire à l'effet sur sa satisfaction de la réaction des autres acteurs en réponse à ses actions ; l'évaluation de la réaction des autres acteurs demande davantage de temps, et augmente donc la durée des simulations.

Dans tous les cas, la valeur du paramètre répartition du renforcement choisie par le modélisateur doit être liée à la structure du modèle, plus précisément à la difficulté de détection de la coopération par les acteurs.

4.6. Ambition Multicritère

Rappelons que selon la SAO, les acteurs d'une organisation sont stratégiques et ajustent leurs comportements de façon finalisée, c'est-à-dire en accord avec une certaine visée, « orientée de façon à atteindre un objectif personnel, compte tenu des contraintes de la situation » ([Friedberg, 1988], cité dans [Scieur, 2005] p. 86). Dans tout ce qui précède, nous avons supposé que l'objectif de l'acteur est de maximiser une certaine grandeur, sa satisfaction, qui est une agrégation des impacts des relations dont il dépend ; cela suppose que ces impacts soient commensurables, que les valeurs s'additionnent linéairement et que celles positives compensent celles négatives. Or il existe des cas où l'acteur poursuivis simultanément plusieurs objectifs indépendants. Ces objectifs sont

incomparables, et une fonction d'agrégation qui attribue une valeur globale permettant d'évaluer la poursuite des objectifs par l'acteur n'a pas de sens. La théorie de l'adaptation de l'aspiration (en anglais, « Aspiration Adaptation Theory »), introduite par [Selten, 1998], prend en considération l'impossibilité de comparer les objectifs hétérogènes, et ne fait pas usage de fonctions d'agrégation des objectifs.

Dans notre contexte, il est possible de représenter l'ambition d'un acteur sous la même forme que sa situation, c'est-à-dire un vecteur d'impact pour chacune des relations dont il dépend. Ceci permet de prendre en compte l'effet spécifique de chaque relation[18] sur un acteur et de ne pas agréger des impacts incommensurables : l'acteur cherche alors à améliorer les impacts obtenues pour chacune des relations au lieu de les sommer dans une grandeur globale, la satisfaction. Le surcroît d'information dont il faut alors doter les acteurs semble réaliste.

Dans le reste de cette section, nous présentons un algorithme de comportement des acteurs qui implémente cette modification de l'algorithme principal (cf. algorithme 3.2). Ensuite nous illustrons l'algorithme sur trois modèles d'organisations.

4.6.1. Présentation de l'algorithme

Cet algorithme de simulation repose sur l'idée que chaque acteur perçoit le détail de sa situation sous la forme d'un vecteur d'impacts des relations dont il dépend, et qu'il cherche à maximiser ces impacts simultanément. Pour chaque relation r, nous avons donc attaché à l'acteur :

- satisfactionR représente l'impact de chaque relation dont il dépend. Elle est mise à jour de la façon suivante :

$$satisfactionR_t = satisfactionR(e_r) = enjeu(r) \times effet_r(e_r) \quad \text{(éq. 4.21)}$$

- *ambitionR* représente son objectif sur chacune des relations dont il dépend. Elle est initialisé à la valeur maximale de l'impact :

$$MaxSatisfactionR = \max_{e_r}(satisfactionR(e_r)) \quad \text{(éq. 4.22)}$$

  Elle est mise à jour de la même façon que dans l'algorithme initiale :

  - si satisfactionR < ambitionR :

$$ambitionR_t = ambitionR_{t-1} - ((1 - txR_{t-1}) \times (réactivité\ /100) \times écartR_t \quad \text{(éq. 4.23)}$$

  - si satisfaction ≥ ambitionR :

$$ambitionR_t = ambitionR_{t-1} + (satisfactionR_t - ambitionR_{t-1}) \times (réactivité\ /100) \quad \text{(éq. 4.24)}$$

- *écartR* correspond à l'écart en proportion entre *satisfactionR* et *ambitionR*. Il est mis à jour de la façon suivante :

$$écartR_t = \frac{ambitionR_{t-1} - satisfactionR_t}{ambitionR_{t-1} - MinSatisfactionR} \quad \text{(éq. 4.25)}$$

  où MinSatisfactionR est l'impact minimal de cette relation sur l'acteur.

$$MinSatisfactionR = \min_{e_r}(satisfactionR(e_r)) \quad \text{(éq. 4.26)}$$

[18] Il est aussi possible de considérer des groupes des relations, donc chacun est constitué de relations commensurables. La situation d'un acteur est représentée alors par un vecteur de l'effet agrégé de chaque groupe de relations, qui est la somme de l'effet de chaque relation de ce groupe, et son objectif est donc de maximiser l'effet de chaque groupe.

- *txR* est le taux d'exploration de l'acteur sur chacune des relations dont il dépend. Il est mis à jour de la façon suivante :

$$txR_t = (1 - (réactivité\ /10)) \times txR_{t-1} + (réactivité\ /10) \times txiR_t \quad \text{(éq. 4.27)}$$

$$txiR_t = 0.1 + (\frac{0.8}{1 + e^{pente \times (écartR_t - abscisse)}}) \quad \text{(éq. 4.28)}$$

où *pente* = – (*ténacité* × (10 – *ténacité*) + 10) et *abscisse* = (10 – *ténacité*) / 10 (éq. 4.29)

Le taux d'exploration global de l'acteur, TX_t, est calculé en fonction des taux d'exploration de chaque relation dont l'acteur dépend de la façon suivante :

$$TX_t = \frac{1}{10} \times \sum_{r \in R\ /\ dépend(a,r)=Vrai} txR_t(r) \times enjeu(a,r) \quad \text{(éq. 4.30)}$$

Cela permet à un acteur de se concentrer sur ses objectifs les plus importants.

- *intensitéR* est l'intensité des actions à réaliser sur cette relation si l'acteur la contrôle. Elle est mise à jour de la façon suivante :

$$\text{intensité}R_t = 2 \times \text{tx}R_t \quad \text{(éq. 4.31)}$$

L'acteur s'estime satisfait si et seulement si chaque composante de son vecteur de satisfaction est plus grande que celle du vecteur d'ambition : Pour chaque relation R, $\text{satisfaction}R_t \geq \text{ambition}R_t$.

Les étapes de l'algorithme sont les suivantes :

Initiation :

Pour chaque relation R :

L'état de R est initialisé arbitrairement à 0 (l'état neutre).

satisfactionR est calculée en fonction de l'état de la relation (éq. 4.21)

ambitionR est initialisé à la valeur maximale de satisfactionR (éq. 4.22)

écartR calculé en fonction de SatisfactionR et ambitionR (éq. 4.25 et 4.26)

txR est initialisé à la valeur du txiR (éq. 4.28 et 4.29)

A chaque étape t de la simulation, chaque acteur :

1. Pour chaque relation r :

a. perçoit sa satisfactionR (éq. 4.21), calcule son écartR (éq. 4.25), et met à jour son ambitionR (éq. 4.23 et 4.24).

b. met à jour son txR (éq. 4.27, 4.28, et 4.29).

c. met à jour l'intensité des actions (équation 4.31).

2. met à jour la force des deux dernières règles appliquées (éq. 4.15 et 4.16), où

$$\Delta satisfaction_t = \sum_{r \in R\ /\ depend(a,r)=Vrai} \Delta satisfactionR_t(r)$$

Les règles de force négative sont oubliées.

3. sélectionne les règles applicables, celles dont la composante situation est proche de sa situation courante en fonction de son discernement (4.2).

4. Si l'ensemble des règles applicables est vide (notamment au début de la simulation), il crée une nouvelle règle, avec une force initialisée à 0, une situation égale à la situation

courante et les actions (modifications à apporter sur les états des relations contrôlées) sont choisies au hasard dans l'intervalle [- intensité ; + intensité] (éq. 4.14).

5. choisit la règle nouvellement créée ou, parmi celles applicables, l'une de trois règles dont la force est la plus grande.

Lorsque tous les acteurs ont choisi une règle, leurs actions sont appliquées.

Algorithme 4.3. *Algorithme Multi-Critère de délibération d'un acteur au cours d'une régulation.*

4.6.2. Exemples d'application

Comme élément de la validation de cet algorithme, nous allons analyser son comportement sur une série de modèles d'organisation ; nous ne prétendrons pas que cette validation soit définitive mais que, sur des tels exemples, il donne les résultats attendus.

Un modèle à deux acteurs et 2n relations

Le modèle d'organisation considéré dans cette section, parfaitement symétrique, a été construit de façon très simple pour permettre une comparaison entre les deux algorithmes. Il comporte deux acteurs et (2*n) relations (cf. tableau 4.14), n allant de 2 jusqu'à 5 (cf. figure 4.51) : chaque acteur contrôle n relations, et ne dépend d'aucune des relations contrôlées par l'autre acteur ; il répartit ses enjeux sur les relations qu'il contrôle (cf. tableau 4.14). Les fonctions d'effets sont linéaires avec une pente de 1 sur les relations dont l'acteur dépend, et nulle sur les autres relations (cf. tableau 4.15). Les résultats des simulations présentées sont réalisés en considérant l'algorithme principal par l'acteur A1, et l'algorithme multi-critère pour A2.

	A1	A2
R11	**5**	0
R12	**5**	0
R21	0	**5**
R22	0	**5**

Tableau 4.14. *Les enjeux des acteurs sur les relations (les contours renforcés indiquent le contrôleur de la relation) pour n = 2.*

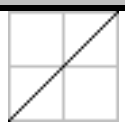
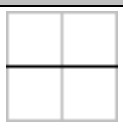
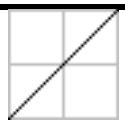
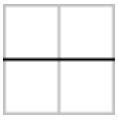
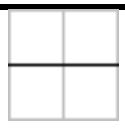
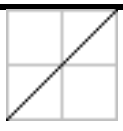
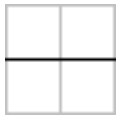
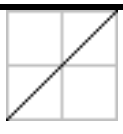
	A1	A2
R11		
R12		
R21		
R22		

Tableau 4.15. *Les fonctions d'effet des relations sur les acteurs.*

La figure 4.51 présente les résultats de 200 simulations. Les paramètres psycho-cognitifs sont initialisés de façon standard : discernement = 1, ténacité = réactivité = 5, et une répartition de 100% pour la dernière règle et 0% pour l'avant-dernière règle.

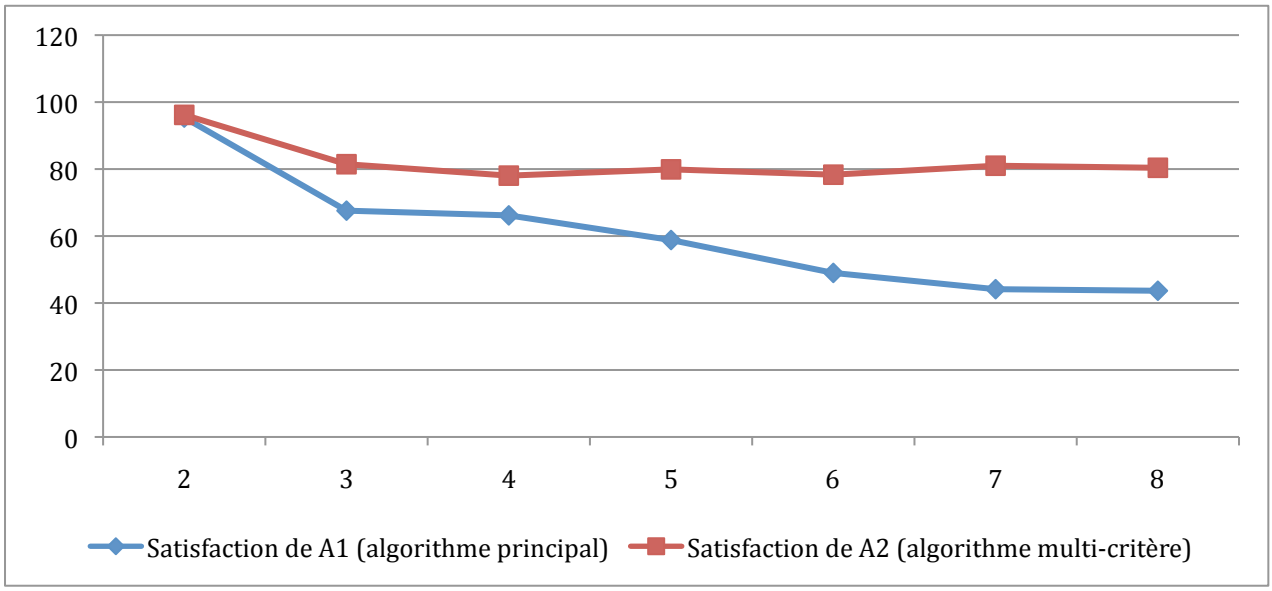


Figure 4.51. *La moyenne de la satisfaction de chaque acteur, en fonction du nombre de relations dont chacun dépend.*

La figure 4.51 montre que, pour A1, plus le nombre des relations dont il dépend augmente, plus sa satisfaction diminue. Tandis que la satisfaction de A2 diminue quand le nombre de relations augmente de 2 à 3, et reste quasiment constante au delà de 3.

En effet, pour n = 2, la satisfaction de A1 dépend de l'impact de chacune des deux relations R21 et R22 ; la diminution de l'une annule l'augmentation de l'autre, A1 doit donc chercher à augmenter simultanément les deux impacts. Pour n = 3, A1 n'est pas capable de détecter si l'augmentation de sa satisfaction provient de l'augmentation de l'impact de chacune des trois relations ou de l'impact des deux relations parmi les trois, et il se contente d'augmenter sa satisfaction sans la comparer avec ce qu'elle pourrait être dans le cas optimal ; plus précisément il se contente d'augmente au moins l'impact de ((n+1) / 2) relations. Plus n augmente, plus l'ambigüité dans le calcul de sa satisfaction augmente, et plus sa satisfaction finale diminue.

Par contre, A2 évalue l'impact de chacune des relations à part ; il fait varier l'ambition locale de chaque relation en fonction de son écart local, et donc de son taux d'exploration local ; il peut explorer sur une des relations, et exploiter sur l'autre. Cela lui permet de mieux analyser sa situation, et d'agir de façon quasiment optimale. En outre, A2, comme A1, doit choisir un vecteur d'actions qui sera évaluée en fonction de son effet sur les impacts des relations, plus précisément comme la somme des variations des impacts. Cela l'empêche de trouver la « meilleure » action pour n > 2, ce qui fait diminuer légèrement sa satisfaction.

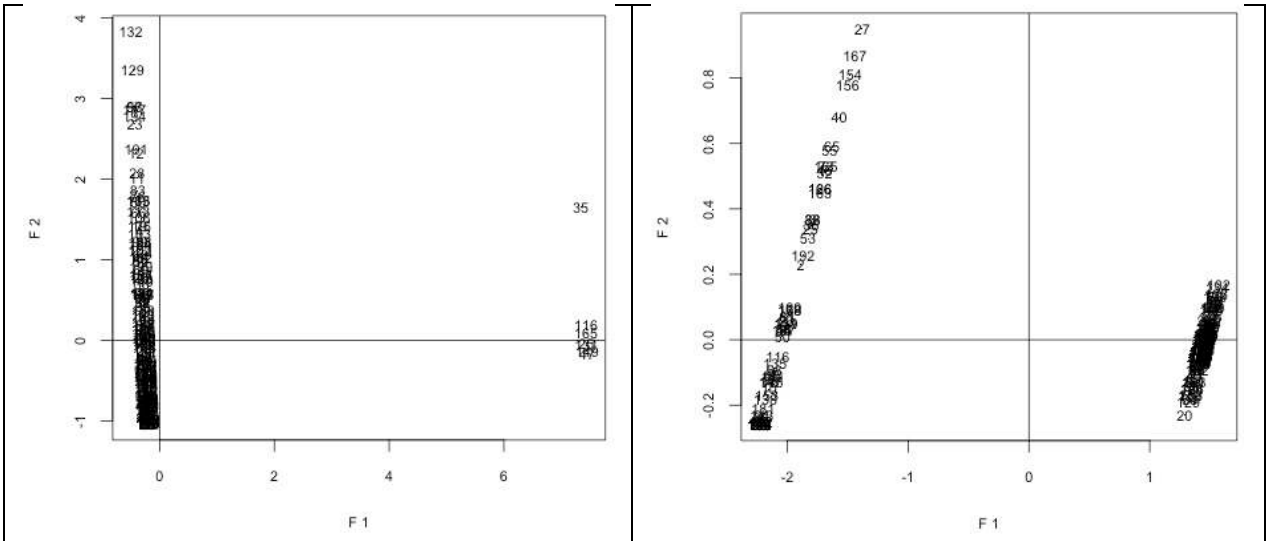
Le dilemme du prisonnier 4/6

Le dilemme du prisonnier traité dans cette analyse est celui avec une répartition 4 / 6. Dans cet exemple, les deux acteurs appliquent l'algorithme Multi-critère. Le tableau 4.16 présente les résultats de 200 simulations avec cet algorithme sur le dilemme du prisonnier 4/6 ainsi que les résultats obtenus avec l'algorithme principal et présentés ci-dessus en 4.4.1. Les paramètres psycho-cognitifs sont initialisés de façon standard : discernement = 1, ténacité = réactivité = 5, et une répartition de 40% pour la dernière règle et 60% pour l'avant-dernière règle. 100% des simulations ont convergé avec une moyenne de 35343 pas nécessaires pour la convergence pour l'algorithme Multi-critère.

		A1	A2	Satisfaction Globale
Multi-critère	Satisfaction	18,6	18,6	37,2
	Satisfaction en %	59,3 %	59,3 %	59,3 %
	Écart-type	2,7	2,7	
Principal	Satisfaction	-3,4	-3,4	-6,8
	Satisfaction en %	48,3%	48,3%	48,3%
	Écart-type	19,42	19,42	

Tableau 4.16. *Satisfaction des acteurs à l'issue de 200 simulations du dilemme du prisonnier 4/6.*

La figure 4.52 montre deux analyses en composantes principales des résultats de simulations du dilemme du prisonnier 4/6 : la figure à gauche correspond aux résultats obtenus avec l'algorithme Multi-Critère, tandis que celle de droite aux résultats obtenus avec l'algorithme principal.



(a) représentation graphique des simulations obtenues avec l'algorithme Multi-critère dans un repère formé des deux axes associés aux deux premières composantes principales (F1 et F2) (la variance est expliquée à 100%).

(b) représentation graphique des simulations obtenues avec l'algorithme principal dans un repère formé des deux axes associés aux deux premières composantes principales (F1 et F2) (100% de la variance est expliquée, dont 98,35 est expliqué par la première composante F1).

Figure 4.52. *Analyse en composantes principales des résultats de simulation du dilemme du prisonnier 4/6 obtenue avec le logiciel R.*

Donnons une interprétation de ces résultats. Le dilemme du prisonnier peut être régulée et donc fonctionner selon deux modalités différentes (cf. figure 4.52) dont la première correspond à l'optima de Pareto qui procure une « bonne » satisfaction, mais pas l'optimale, pour les deux acteurs, tandis que la deuxième correspond à l'équilibre de Nash et procure une « faible » satisfaction, moins bonne que la première mais loin d'être le pire, pour les deux acteurs. Ce sont toujours les mêmes états qui sont obtenus[19], cela tient à la structure très simple du modèle. A cause

[19] Cette analyse tient en compte le nombre de pas nécessaire pour la convergence qui varie d'une simulation à une autre, c'est pour cette raison que les points dans les deux figures de 4.17 ne sont pas juxtaposés.

de la grande imprécision des informations sur les effets de leurs actions et de leur faible ténacité, les deux acteurs ne sont pas capables de détecter toujours l'intérêt de la coopération, ils jouent, dans environ 57% des simulations, de façon défensive et se contentent d'assurer une « faible » satisfaction sans chercher à l'améliorer.

L'algorithme Multi-critère permet à un acteur de repérer les relations les plus importantes eu égard à ses objectifs, et de se concentrer sur l'amélioration de l'impact de ces relations. Cette importance est définie par les enjeux posés par l'acteur sur les relations (cf. éq. 4.30). Dans le dilemme du prisonnier considéré dans cette analyse, l'acteur aperçoit, dans environ 96% de simulations, que la relation contrôlée par l'autre acteur est plus profitable que celle qu'il contrôle, et cherche donc à maximiser l'effet de cette relation sur lui-même en coopérant avec l'autre acteur. Les satisfactions des acteurs obtenues sont proches de l'optima de Pareto (cf. tableau 4.16), et les simulations convergent donc très majoritairement vers la modalité la plus bénéfique (cf. figure 4.52.a).

Le modèle free-rider

Le cas considéré dans ce paragraphe est le modèle free-rider où les quatre acteurs appliquent l'algorithme Multi-critère. Le tableau 4.17 présente, dans la première ligne, les résultats sur 200 simulations, et dans la dernière ligne, les résultats obtenus précédemment en simulant ce modèle avec l'algorithme principal (cf. 4.4.3). Les paramètres psycho-cognitifs sont initialisés de façon standard : discernement = 1, ténacité = réactivité = 5, et une répartition de 10% pour la dernière règle et 90% pour l'avant-dernière règle.

	Configurations	C1	C2	C3	C4	C5	C6	C7	C8	C9
Algo. multi-critère	% d'occurrences	11,5	27,5	32	29	0	0	0	0	0
Algo. principal		10,5	29,5	30,5	29,5	0	0	0	0	0

Tableau 4.17. *Le pourcentage d'apparition de chacun des neuf états du système dans les résultats de la simulation : l'état qui maximise la satisfaction globale (C1) ; les quatre états qui maximisent la satisfaction de l'un des quatre acteurs : A1 (C8), A2 (C2), A3 (C3) et A4 (C4) ; trois états dans lesquels deux des trois acteurs ne coopèrent pas : A2 et A3 (C5), A2 et A4 (C6), A3 et A4 (C7) ; l'équilibre de Nash (C9).*

Les résultats obtenus avec l'algorithme Multi-critère, sont quasiment les mêmes que ceux obtenus avec l'algorithme principal. En effet, le fait que A1 contrôle une seule relation le rend incapable de réagir à la trahison d'un acteur parmi les trois autres acteurs, bien qu'il évalue sa situation de façon beaucoup plus précise; s'il décidait de ne pas coopérer, les autres auraient une satisfaction très basse ce qu'ils ne peuvent accepter. En général, il ne faut pas s'attendre à ce que l'acteur arrive toujours à augmenter sa satisfaction lorsqu'il considère l'impact de chaque relation indépendamment des autres. Cela dépend de sa capacité à prendre en considération cette « meilleure » évaluation et de réagir en conséquence.

Chapitre 5 – Variations sur la rationalité des acteurs

Nous avons présenté dans le chapitre précédent un modèle de la rationalité d'acteurs sociaux exclusivement intéressés à la maximisation de leur intérêt, défini par la structure du jeu social comme leur *satisfaction*, sous la contrainte d'une rationalité limitée et comportant un léger biais en faveur de l'atteinte d'un compromis entre les intérêts, par nature conflictuels, des acteurs, justifié par le fait qu'un tel compromis est indispensable à la pérennité de l'organisation à laquelle l'acteur a intérêt.

Ce modèle de rationalité des acteurs sociaux est minimaliste en ce qui concerne la connaissance dont chaque acteur dispose pour déterminer son comportement : la valeur courante de sa satisfaction, de façon agrégée et sans connaissance de la façon dont cette valeur est obtenue, et, pour l'initialisation de l'algorithme, la valeur maximale que cette satisfaction peut atteindre pour une certaine configuration de l'organisation.

De par la nature même des organisations sociales, la maximisation de l'intérêt des acteurs passe, globalement, par leur coopération, du moins dans la mesure où elle ne leur est pas trop coûteuse. Dans l'ensemble, c'est bien vers ce type de configurations que cet algorithme converge : vers des configurations que l'on peut qualifier de Pareto-équitable[20], dans la mesure où elles correspondent à des optima de Pareto tout en écartant les configurations très défavorables à l'un des acteurs. Par exemple, l'algorithme appliqué au modèle free-rider présenté en 2.2.3 converge vers les configurations C1, C2, C3 ou C4 mais jamais vers les configurations C5, C6, C7 ou C8 (cf. 4.4.3).

Il existe bien d'autres configurations d'une organisation qui satisfont une propriété remarquable, au premier chef desquelles les configurations qui correspondent au maximum ou au minimum de la satisfaction de l'un des acteurs, ou bien au maximum ou au minimum de la somme des satisfactions d'un quelconque sous-ensemble d'acteurs de l'organisation. Dans ce chapitre, nous allons proposer des modèles pour des rationalités qui, lorsqu'elles sont adoptées par les acteurs de l'organisation, convergent vers des configurations autres que celles Pareto-équitables. Plus précisément, nous allons considérer des rationalités des acteurs qui conduiraient une organisation de se réguler dans une configuration proche de l'équilibre de Nash, une configuration qui maximise la satisfaction globale (*i.e.* la somme des satisfactions de tous les acteurs), une configuration élitiste qui maximise (ou à l'inverse anti-élitiste qui minimise) la satisfaction de l'acteur le plus satisfait, une configuration protectrice qui maximise (ou à l'inverse anti-protectrice qui minimise) la satisfaction de l'acteur le moins satisfait, une configuration égalitariste qui minimise (ou à l'inverse anti-égalitariste qui maximise) l'écart entre la satisfaction de l'acteur le plus satisfait et celle de l'acteur le moins satisfait.

Les modèles de rationalité des acteurs introduits dans ce chapitre restent cognitivement et socialement vraisemblables. L'algorithme de simulation de chaque type de rationalité résulte de modifications apportées à l'algorithme principal présenté dans le chapitre 4.

Dans un premier temps, nous présentons un algorithme de simulation qui correspond à un acteur agissant de façon purement individualiste et défensive, et conduit l'organisation à se réguler vers une configuration correspondant à un équilibre de Nash. Cet algorithme est présenté dans la section 5.1.

Ensuite, nous présentons des variations sur l'individualisme d'un acteur qui le fait coordonner ses actions en faveur de l'organisation (5.2), ou bien se focaliser sur la situation d'un ou de deux

[20] Nous employons le terme Pareto-équitable plutôt que Pareto-égalitaire, car la distribution des satisfactions opérée par une configuration doit principalement s'évaluer en proportion, c'est-à-dire considérer la satisfaction que l'acteur obtient par rapport à celle qu'il pourrait obtenir, et non pas seulement en valeur.

autre(s) acteur(s) (5.3, 5.4, et 5.5). Pour ce faire, nous introduisons un paramètre psycho-social (cf. 5.2.1), l'*égocentrisme (EC)*, permettant au modélisateur de spécifier à quel point un acteur est prêt à évaluer et à prendre en considération la situation de certains autres.

Pour chaque type de rationalité, nous présentons un algorithme du comportement des acteurs qui est une variante de l'algorithme présenté dans le chapitre 4. Ensuite nous illustrons l'algorithme sur des organisations formelles, essentiellement le dilemme du prisonnier et le modèle free-rider, et sur des organisations réelles, le cas Bolet et le cas Seita.

5.1 – L'équilibre de Nash

Dans une configuration qui est un équilibre de Nash [Nash, 1951], aucun acteur ne peut modifier seul sa stratégie sans affaiblir sa situation. Chaque acteur joue défensif de façon à obtenir le maximum de ce qu'il peut s'assurer d'obtenir, y compris dans le cas du comportement des autres acteurs qui soit le pire pour lui. Cette stratégie est surtout pertinente et utilisée dans les jeux essentiellement compétitifs où les acteurs ont des objectifs opposés, et chacun peut s'intéresser à réaliser son objectif sans se préoccuper de ce qu'il en est pour les autres acteurs.

De par le méta-modèle de la structure des organisations de SocLab, toute organisation a au moins un équilibre de Nash, et il peut être calculé analytiquement : c'est une configuration telle que chaque relation est dans un état qui maximise son effet sur l'acteur qui la contrôle. Dans un tel état, tout acteur qui est le seul à modifier son comportement voit sa satisfaction diminuer. Réciproquement, si toute modification de l'état d'une relation diminue la satisfaction de l'acteur qui la contrôle, c'est que cette relation est dans un état dont l'effet sur cet acteur est maximal et donc correspond à un équilibre de Nash.

5.1.1 – Présentation de l'algorithme

Cet algorithme, que l'on nomme *algorithme de Nash*, repose sur l'évaluation, par chaque acteur, de sa situation par les impacts sur lui-même des relations qu'il contrôle, variable que nous appellerons son *Auto-satisfaction*. Cette variable diffère de la *satisfaction* (présentée dans le chapitre 4) du fait qu'elle ne prend pas en considération les impacts sur l'acteur des relations contrôlées par les autres.

$$autoSatisfaction_t = autoSatisfaction(e) = \sum_{r \in R \,/\, contrôle(r)=a} enjeu(r) \times effet_r(e_r) \qquad \text{(éq. 5.1)}$$

A chaque étape, l'acteur perçoit son auto-satisfaction, et c'est la grandeur qu'il vise à maximiser. Tous les paramètres et les variables seront initialisés et mis à jour de la même façon que dans l'algorithme principal, mais en remplaçant la satisfaction de l'acteur par son auto-satisfaction, de la façon suivante :

- L'ambition d'un acteur est initialisée à son auto-satisfaction maximale calculée de la façon suivante :

$$ambition_0 = \sum_{r \in R \,/\, contrôle(r)=a} \max_{e_r}(enjeu(r) \times effet_r(e_r)) \qquad \text{(éq. 5.2)}$$

  La mise à jour de l'ambition diffère de celle de l'algorithme principal du fait de l'utilisation de l'auto-satisfaction au lieu de la satisfaction, et est faite de la façon suivante :

  - Si auto-satisfaction < ambition :

$$ambition_t = ambition_{t-1} - ((1 - TX_{t-1}) \times (réactivité / 10^5) \times écart_t) \qquad \text{(éq. 5.3)}$$

- Si auto-satisfaction ≥ ambition :

$$ambition_t = ambition_{t-1} + ((autoSatisfaction_t - ambition_{t-1}) \times (réactivité / 10^2)) \quad \text{(éq. 5.4)}$$

Dans cet l'algorithme, l'ambition diminue beaucoup moins vite que dans l'algorithme principal : le dénominateur de l'équation 5.3 est 10^5 comparé à 10^2 dans l'algorithme principal. En effet, il faut tenir compte du fait qu'il existe une configuration, celle que l'on cherche à atteindre, dans laquelle chaque acteur atteint effectivement son ambition initiale, ce qui n'est pas le cas pour la plupart des organisations avec l'algorithme principal.

- La mise à jour de l'écart se fait de la même façon que dans l'algorithme principal, mais en considérant l'auto-satisfaction au lieu de la satisfaction :

$$écart_t = \frac{ambition_{t-1} - autoSatisfaction_t}{ambition_{t-1} - MinAutoSatisfaction} \quad \text{(éq. 5.5)}$$

où MinAutoSatisfaction est l'auto-satisfaction minimale calculée de la façon suivante :

$$MinAutoSatisfaction = \sum_{r \in R / contrôle(r)=a} \min_{e_r}(enjeu(r) \times effet_r(e_r)) \quad \text{(éq. 5.6)}$$

- La mise à jour des forces des règles diffère de celle dans l'algorithme principal du fait que l'acteur perçoit immédiatement l'effet de ses actions, et il n'a plus besoin d'attendre à l'étape t+2 pour effectuer une évaluation complète de ses actions. Cette mise à jour se fait donc uniquement à l'étape t+1 de la façon suivante :

$$RA_{t-1}.force_t = (1 - TX_t) \times RA_{t-1}.force_{t-1} + TX_t \times \Delta autoSatisfaction_t \quad \text{(éq. 5.7)}$$

$$\text{où } \Delta autoSatisfaction_t = autoSatisfaction_t - autoSatisfaction_{t-1} \quad \text{(éq. 5.8)}$$

A noter que si ΔautoSatisfaction est nulle et l'acteur n'est pas encore satisfait, la règle est oubliée puisque chaque acteur ne cherche pas une « bonne » situation comme dans l'algorithme principal, mais à atteindre son ambition initiale, qu'il sait être accessible. Si une règle ne lui permet plus d'augmenter son auto-satisfaction et qu'il n'est pas encore satisfait, il la considère comme « mauvaise » car elle ne lui permet pas d'atteindre son objectif principal, et elle sera donc oubliée.

- L'acteur s'estime satisfait si son auto-satisfaction est plus grande ou égale à son ambition.

Pour résumer, cet algorithme diffère de l'algorithme principal dans l'étape de l'initialisation des variables, et dans les étapes 1 et 4 : c'est l'auto-satisfaction d'un acteur qui est évaluée au lieu de sa satisfaction, et est utilisée dans la mise à jour de son écart et de son ambition. De plus, les règles appliquées sont évaluées uniquement à l'étape suivante. Les étapes de cet algorithme sont présentées de la façon suivante :

Initialisation :

L'état de chaque relation est initialisé arbitrairement à 0 (l'état neutre).

L'auto-satisfaction est calculée en fonction des états des seules relations que l'acteur contrôle (éq. 5.1)

L'ambition est initialisée à la valeur maximale de l'auto-satisfaction de l'acteur (éq. 5.2)

L'écart est calculé en fonction de l'auto-satisfaction et l'ambition (éq. 5.5 et 5.6)

Le taux d'exploration est initialisé à la valeur du taux d'exploration instantanée (éq. 4.9, 4.10, et 4.11)

Répéter

A chaque étape t de la simulation, chaque acteur :

1. perçoit son auto-satisfaction (éq. 5.1), calcule son écart (éq. 5.5), et met à jour son ambition (éq. 5.3 et 5.4).

2. met à jour son taux d'exploration (éq. 4.9, 4.10, 4.11, et 4.12).

3. met à jour l'intensité des actions (équation 4.13).

4. met à jour la force de la règle appliquée (éq. 5.7 et 5.8). Les règles de force négative sont oubliées.

5. sélectionne les règles applicables, celles dont la composante situation est proche de sa situation courante en fonction de son discernement (4.2).

6. Si l'ensemble des règles applicables est vide (notamment au début de la simulation), il crée une nouvelle règle, avec une force initialisée à 0, une situation égale à la situation courante et les actions choisies au hasard dans l'intervalle [- intensité ; + intensité] (éq. 4.14).

7. choisit la règle nouvellement créée ou, parmi celles applicables, l'une des règles dont la force est la plus grande.

Lorsque tous les acteurs ont choisi une règle, leurs actions sont appliquées.

Jusqu'à (auto-satisfaction >= ambition) pour chacun des acteurs

Algorithme 5.1. *Schéma de l'algorithme de rationalité des acteurs qui régule une organisation dans une configuration proche de l'équilibre de Nash.*

Cet algorithme suppose que l'acteur est capable de distinguer précisément, dans les satisfactions qu'il reçoit, ce qui dépend de lui et ce qui dépend des autres, sans pour autant avoir à faire le détail des relations dont il dépend. On peut considérer qu'il est assez vraisemblable que cette information soit perceptible par la plupart des acteurs sociaux.

5.1.2 – Exemples d'application

Afin d'évaluer la qualité de cet algorithme, nous avons analysé son comportement sur plusieurs modèles d'organisations formelles, notamment sur le dilemme du prisonnier pour les différentes répartitions des enjeux (cf. 4.4.1), sur les modèles du dilemme du prisonnier à n-acteurs, et sur le modèle free-rider. Les résultats de simulations sont satisfaisants : une convergence très rapide avec 50 pas en moyenne, et l'état final obtenu à la fin des simulations pour chacun de ces modèles est quasiment égal à l'équilibre de Nash du modèle.

Dans cette section, nous présentons les résultats des analyses sur deux modèles d'organisations : le cas Bolet et un modèle à deux acteurs et huit relations.

Le cas Bolet

Le tableau 5.1 présente les résultats de 200 simulations de cet algorithme sur le cas Bolet et les satisfactions des quatre acteurs à l'équilibre de Nash dans la dernière ligne. Les paramètres psycho-cognitifs sont initialisés de façon standard : discernement = 1, ténacité = réactivité = 5. Toutes les simulations ont convergé en moins de 2 000 pas.

		CA	Père	André	Jean-BE
Algorithme de Nash	auto-satisfaction	30	4	25,5	19,3
	Satisfaction	-12,71	56,23	40,03	45,98
	Ecart-Type	0,92	0,08	0	1,01
Satisfactions à l'équilibre de Nash		-12,7	56,2	40,1	46

Tableau 5.1. *Satisfaction des acteurs à l'issue de 200 simulations du cas Bolet avec l'algorithme de Nash ainsi que celles à l'équilibre de Nash.*

Le cas Bolet présente une diversité de fonctions d'effets ; les acteurs dépendent de certaines relations par des fonctions d'effet linéaires, mais dépendent aussi d'autres par des fonctions sigmoïdes ou quadratiques. Les fonctions quadratiques n'étant pas monotones, leurs extrema ne coïncident pas avec les bornes inférieures ou supérieures des intervalles des états des relations. Les résultats présentés ci-dessus montre que notre algorithme est capable de gérer cette diversité, et donne des résultats très satisfaisants.

Un modèle à deux acteurs et huit relations

Le modèle d'organisation traité dans cette section a été construit pour montrer l'efficacité de l'algorithme de Nash sur les modèles dans lesquels chaque acteur contrôle plus d'une seule relation. Il comporte deux acteurs et huit relations (cf. tableau 5.2). Les enjeux des acteurs sur les relations et les fonctions d'effet sont définis comme montrés dans les tableaux 5.2 et 5.3.

	A1	A2
R11	**1,5**	1,5
R12	**1,5**	1,5
R13	**1,5**	0,5
R14	**1,5**	0,5
R21	1,5	**1,5**
R22	1,5	**1,5**
R23	0,5	**1,5**
R24	0,5	**1,5**

Tableau 5.2. *Les enjeux des acteurs sur les relations (les contours renforcés indiquent le contrôleur de la relation).*

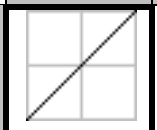
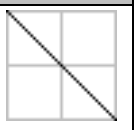
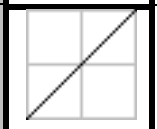
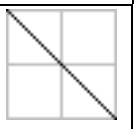
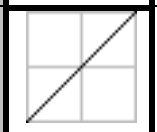
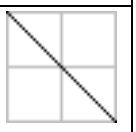
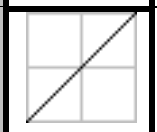
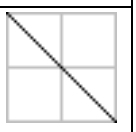
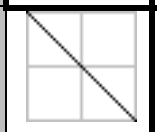
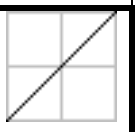
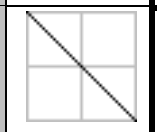
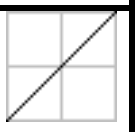
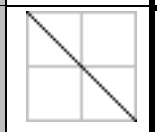
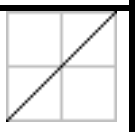
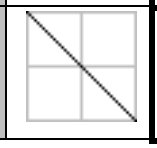
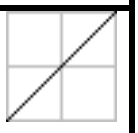
	A1	A2
R11		
R12		
R13		
R14		
R21		
R22		
R23		
R24		

Tableau 5.3. *Les fonctions d'effet des relations sur les acteurs.*

Le tableau 5.4 présente les résultats de 200 simulations de cet algorithme sur ce modèle et les satisfactions des deux acteurs à l'équilibre de Nash dans la dernière ligne. Les paramètres psycho-cognitifs sont initialisés de façon standard : discernement = 1, ténacité = réactivité = 5. 100% des simulations ont convergé en moins de 400 pas.

		A1	A2
Algorithme de Nash	auto-satisfaction	60	60
	Satisfaction	20	20
	Ecart-Type	0	0
Satisfactions à l'équilibre de Nash		20	20

Tableau 5.4. *Satisfaction des acteurs à l'issue de 200 simulations d'un modèle à deux acteurs et huit relations avec l'algorithme de Nash (trois premières lignes) ainsi que celles à l'équilibre de Nash.*

Ce modèle présente une dispersion du contrôle exercé par chaque acteur sur quatre relations. Cette dispersion rend l'équilibre de Nash difficile à trouver puisque l'acteur calcule son auto-satisfaction de façon agrégée en sommant les impacts des relations qu'il contrôle, et l'augmentation de l'impact d'une relation peut compenser la diminution de l'impact d'une autre relation : plus le

nombre des relations augmente, plus cette ambigüité dans le calcul de son auto-satisfaction augmente, et plus l'équilibre est difficile à trouver. Les résultats présentés dans le tableau 5.4 montrent que notre algorithme est aussi capable de gérer cette dispersion, et donne exactement les résultats attendus.

5.1.3 – Discussion

Les deux exemples utilisés ci-dessus montrent que cet algorithme converge de façon beaucoup plus rapide et précise vers les résultats attendus que l'algorithme principal étudié au chapitre précédent. Cela tient à ce que les règles soient renforcées exclusivement en fonction de leur effet sur la grandeur que l'on cherche à optimiser. A contrario, cela met en évidence à quel point, dans l'algorithme principal, les acteurs disposent de peu d'information sur l'état du jeu : l'effet de l'action d'une règle entre pour assez peu dans le différentiel de satisfaction qui sera utilisé pour évaluer la qualité de cette règle.

5.2 – L'optimisation de la satisfaction globale

Dans les jeux sociaux, chaque acteur cherche à réaliser son objectif personnel tout en tenant compte de l'objectif global du système. Considérons par exemple une équipe de foot constituée de plusieurs joueurs : chacun des joueurs, quelque soit sa position, cherche à montrer ses capacités personnelles dans le jeu afin d'améliorer sa réputation, mais il cherche aussi à participer à la réalisation de l'objectif global de l'équipe : gagner le match.

Dans notre contexte, nous pouvons définir l'objectif global d'une organisation comme la maximisation de sa *satisfaction globale*. Cette grandeur est évaluée comme la somme des satisfactions de tous les acteurs de l'organisation, ou par tout autre opérateur d'agrégation[21]. Chacun des acteurs cherche donc à réaliser un objectif qui résulte d'une combinaison entre son objectif personnel et l'objectif global de l'organisation à proportion de son égocentrisme. Ce paramètre détermine à quel point un acteur est prêt à négliger son objectif personnel en faveur de celui de l'organisation.

Avant d'entamer dans la présentation de cet algorithme (5.2.2) et l'analyse des résultats de simulations (5.2.3) et de sensibilité (5.2.4) sur différents modèles d'organisation, nous introduisons le paramètre psycho-social, l'égocentrisme d'un acteur, qui sera aussi utilisé dans tous les algorithmes suivants.

5.2.1 – L'égocentrisme

Contrairement à l'algorithme principal, nous considérons dans cet algorithme, et dans tous les algorithmes suivants, que chaque acteur a deux objectifs : un objectif personnel qui consiste à maximiser sa propre satisfaction, et un objectif social dont la nature dépend de la rationalité de l'acteur. Par exemple, un acteur élitiste (5.3) a un objectif social de soutenir la satisfaction de l'acteur ayant la satisfaction maximale.

Le paramètre égocentrisme déterminer l'importance relative que l'acteur accorde à son objectif personnel et à son objectif social dans son processus d'apprentissage. En d'autres termes, il détermine à quel point un acteur est prêt à négliger son objectif personnel en faveur de celui social. Il intervient dans le calcul de la variation de sa satisfaction et de son écart (par exemple, cf. 5.10 et 5.11). Plus l'égocentrisme d'un acteur diminue, plus la proportion de son objectif social augmente et celle de son objectif personnel diminue.

[21] Il est aussi possible de considérer, dans certaines organisations, que l'objectif global de l'organisation est porté par l'un de ses membres, ou quelques uns de ses membres.

L'échelle de valeur de l'égocentrisme d'un acteur varie entre 0,1 et 1. La valeur 0 est exclue car elle correspond à un acteur qui ne s'intéresse pas du tout à sa propre situation ce qui est un comportement plutôt pathologique. La valeur 1 correspond à un acteur purement individuel qui ne s'intéresse qu'à sa propre satisfaction ; on se retrouve quasiment avec l'algorithme principal présenté au chapitre 4. En effet, pour un égocentrisme de 1, l'acteur agit de façon individuel et ne participe pas à la réalisation de son objectif social, mais il ne s'estime pas satisfait tant que cet objectif n'est pas réalisé : par exemple, un acteur protecteur, quelque soit la valeur de son égocentrisme, n'est pas satisfait tant que l'acteur ayant la satisfaction minimale n'est pas satisfait. Une grande valeur de ce paramètre (0,9) correspond à un acteur qui attache une forte importance à sa propre situation et peu d'importance aux autres. Tandis qu'une faible valeur (0,1) correspond à un acteur qui accorde davantage d'attention aux autres qu'à lui-même.

5.2.2 – Présentation de l'algorithme

Cet algorithme, que l'on nomme *algorithme de l'objectif global*, repose sur l'évaluation, par chaque acteur, de la situation courante de l'organisation, plus précisément de la progression de la satisfaction de l'organisation, que nous appellerons Δ*satisfactionGlobale*. Cette variable est calculée comme la somme de la variation des satisfactions de tous les acteurs, ou par tout autre façon d'agréger ces variations :

$$\Delta satisfactionGlobale_t = \sum_{a \in A} \Delta satisfaction_t(a) \qquad \text{(éq. 5.9)}$$

La variation de la satisfaction de l'acteur est calculée en fonction de la variation de sa satisfaction personnelle, ΔsatisPerso (cf. éq. 4.17), et de celle de la satisfaction globale de la façon suivante :

$$\Delta satisfaction_t = EC \times \Delta satisPerso_t + (1 - EC) \times \Delta satisfactionGlobale_t \qquad \text{(éq. 5.10)}$$

où EC est l'égocentrisme de l'acteur.

De plus, chaque acteur calcule son écart à chaque instant en fonction de son écart personnel (cf. éq. 4.4) et de celui de l'organisation (cf. éq. 5.15) de la façon suivante :

$$écart_t = EC \times écartPerso_t + (1 - EC) \times écartOrg_t \qquad \text{(éq. 5.11)}$$

$$\text{où } écartOrg_t = (satisOrgMax - satisOrg_t) / (satisOrgMax - satisOrgMin) \qquad \text{(éq. 5.12)}$$

et satisOrgMax (respectivement satisOrgMin) est la satisfaction globale maximale (respectivement minimale) de l'organisation, et $satisOrg_t$ est la satisfaction de l'organisation à un instant t, égale à la somme des satisfactions de tous les acteurs.

Les autres variables de cet algorithme (l'ambition, le taux d'exploration, l'intensité des actions, et les forces des règles) sont calculées et mises à jour de la même façon que dans l'algorithme principal. L'acteur est satisfait si et seulement si sa satisfaction est plus grande que son ambition.

Cet algorithme diffère de l'algorithme principal dans les étapes 1 et 4 : l'écart d'un acteur est une combinaison entre son écart personnel et l'écart de l'organisation, et il en est de même pour la variation de sa satisfaction. Les étapes de cet algorithme sont présentées de la façon suivante :

Initialisation :

L'état de chaque relation est initialisé arbitrairement à 0 (l'état neutre).

La satisfaction est calculée en fonction des états des relations dont l'acteur dépend (éq. 4.3)

L'ambition est initialisée à la valeur maximale de la satisfaction de l'acteur (éq. 4.6)

L'écart est calculé en fonction de la satisfaction et l'ambition (éq. 4.4 et 4.5)

Le taux d'exploration est initialisé à la valeur du taux d'exploration instantané (éq. 4.9, 4.10, et 4.11)

Répéter

A chaque étape t de la simulation, chaque acteur :

1. perçoit sa satisfaction (éq. 4.3), calcule son écart (éq. 4.4, 5.11 et 5.12), et met à jour son ambition (éq. 4.7 et 4.8).

2. met à jour son taux d'exploration (éq. 4.9, 4.10, 4.11, et 4.12).

3. met à jour l'intensité des actions (équation 4.13).

4. met à jour la force des deux dernières règles appliquées (éq. 4.15, 4.16, 4.17, 5.9 et 5.10). Les règles de force négative sont oubliées.

5. sélectionne les règles applicables, celles dont la composante situation est proche de sa situation courante en fonction de son discernement (4.2).

6. Si l'ensemble des règles applicables est vide (notamment au début de la simulation), il crée une nouvelle règle, avec une force initialisée à 0, une situation égale à la situation courante et les actions (modifications à apporter sur les états des relations contrôlées) choisies au hasard dans l'intervalle [- intensité ; + intensité] (éq. 4.14).

7. choisit la règle nouvellement créée ou, parmi celles applicables, l'une de trois règles dont la force est la plus grande.

Lorsque tous les acteurs ont choisi une règle, leurs actions sont appliquées.

Jusqu'à (satisfaction >= ambition) pour chacun des acteurs

Algorithme 5.2. *Schéma de l'algorithme de rationalité des acteurs qui régule une organisation dans une configuration proche de l'état maximisant la satisfaction globale.*

Cet algorithme suppose que l'acteur est capable de percevoir globalement le plus ou moins bon fonctionnement de l'organisation à laquelle il appartient, ce qui est tout à fait vraisemblable. Cette connaissance est du même ordre que celle qui permet à un acteur d'exercer des solidarités.

5.2.3 – Exemples d'application

Afin d'évaluer la qualité de cet algorithme, nous avons analysé son comportement sur plusieurs modèles d'organisations formelles, notamment sur le dilemme du prisonnier pour les différentes répartitions des enjeux (cf. 4.4.1), et sur les modèles du dilemme du prisonnier à n-acteurs. Les résultats de simulations sont satisfaisants : une convergence rapide avec une moyenne maximale de 13 000 pas (sauf pour le dilemme du prisonnier à 5 acteurs où cette durée augmente jusqu'à 70 000), et la satisfaction globale obtenue à la fin des simulations de cet algorithme sur chaque modèle est quasiment égale sa valeur maximale.

Dans cette section, nous présentons les résultats des analyses sur deux autres modèles d'organisations, le cas Bolet et le modèle free-rider.

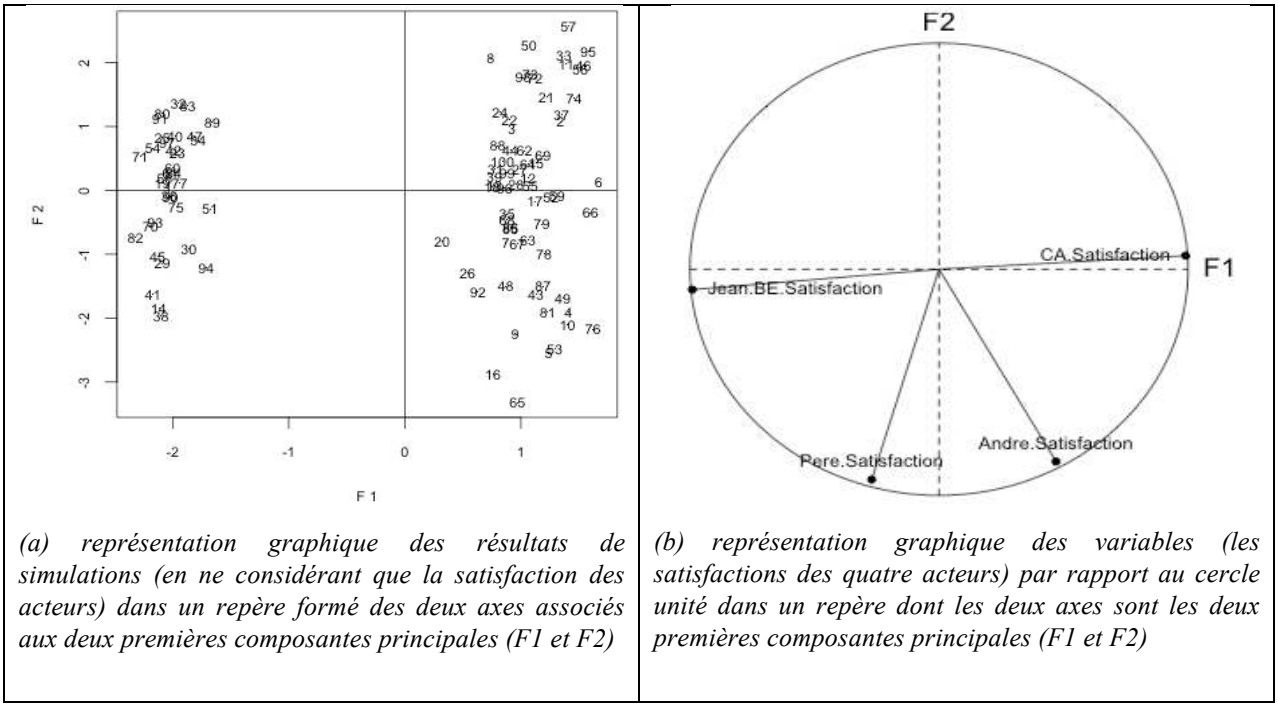
Le cas Bolet

Le tableau 5.5 présente les résultats de 200 simulations du cas Bolet avec cet algorithme, les résultats obtenus avec l'algorithme principal présentés dans 4.4.4 ainsi que, dans la dernière ligne, les satisfactions des quatre acteurs et la satisfaction globale dans l'état qui maximise la satisfaction globale. Les paramètres psycho-cognitifs et psycho-social sont initialisés de façon standard : discernement = 1, ténacité = réactivité = 5, égocentrisme = 0,5, et une répartition de 10% pour la dernière règle et 90% pour l'avant dernière règle. Toutes les simulations ont convergé en moins de 130 000 pas avec une moyenne de 36 800.

		CA	Père	André	Jean-BE	Satisfaction Globale
Algorithme de l'objectif global	Satisfaction	34,78	57,32	51,52	20,32	163,94
	Satisfaction en %	67,39 %	93,12 %	81,25 %	62,53 %	90,01 %
	Ecart-Type	32,92	1,48	6,14	20,67	
Algorithme Principal	Satisfaction	-12,03	56,98	43,61	46,97	135,53
	Satisfaction en %	43,99 %	92,85 %	76,11 %	78,96 %	82,77 %
	Ecart-Type	3,19	1,08	3,38	3,49	
Satisfaction Globale Maximale		63,3	62,9	74,8	2,2	**203,2**

Tableau 5.5. *Satisfaction des acteurs à l'issue de 200 simulations du cas Bolet avec l'algorithme de l'objectif global et avec l'algorithme principal ainsi que, dans la dernière ligne, les satisfactions dans l'état qui maximise la satisfaction globale.*

La figure 5.1 montre une analyse en composantes principales des résultats de simulations du cas Bolet obtenus avec l'algorithme de l'objectif global qui explique l'écart type très important de CA et de Jean-BE : l'algorithme ne disperse pas aléatoirement les configurations qu'il calcule, mais il les répartit en deux modes bien distincts.



(a) représentation graphique des résultats de simulations (en ne considérant que la satisfaction des acteurs) dans un repère formé des deux axes associés aux deux premières composantes principales (F1 et F2)

(b) représentation graphique des variables (les satisfactions des quatre acteurs) par rapport au cercle unité dans un repère dont les deux axes sont les deux premières composantes principales (F1 et F2)

Figure 5.1. *Analyse en composantes principales des résultats de simulation du cas Bolet obtenue avec le logiciel R (96,09 de la variance est expliquée).*

Donnons une interprétation à ces résultats. Le cas Bolet présente un conflit structurel entre le chef d'atelier et Jean-BE (cf. figure 5.1.b). Avec l'algorithme principal, Jean-BE, ayant le support des deux autres acteurs, a réussi à tourner ce conflit en sa faveur (cf. 4.4.4). Inversement, avec l'algorithme de l'objectif global, chaque acteur poursuit aussi l'objectif social consistant à maximiser la satisfaction globale de l'organisation ; il participe donc plus ou moins à la réalisation de cet objectif en fonction de son égocentrisme qui est fixé dans cette analyse à 0,5 pour chacun des quatre acteurs. L'état qui maximise la satisfaction globale est largement au bénéfice du chef d'atelier, et au détriment du Jean-BE (cf. la dernière ligne du tableau 5.5), ce qui n'est pas acceptable dans tous les cas par Jean-BE qui dispose du pouvoir d'en dissuader les autres acteurs. La figure 5.1.a montre qu'il existe deux modes de fonctionnement de cette organisation : le premier, qui conduit à une satisfaction globale proche de l'optimal, est au bénéfice du chef d'atelier et est atteint pour environ 60% de simulations, tandis que le deuxième, qui conduit à une satisfaction globale moins bonne que celle du premier, est au bénéfice de Jean-BE et est atteint pour environ 40% de simulations.

Une Classification Ascendante Hiérarchique (cf. chapitre 4.3.2) permet de bien caractériser les deux modes vers lesquels les simulations convergent. Le tableau 5.6 montre la moyenne de l'état des relations et de la satisfaction des acteurs pour chacun des modes, et les figures 5.2 et 5.3 les box plots des relations et des acteurs dans chacun des deux modes. Il apparaît clairement que ce sont l'état des relations 'décision d'achat' et 'application de la prescription' qui varient principalement entre ces deux modes, et qui sont à l'origine des variations de satisfaction de CA et Jean-BE.

		Mode 1	Mode 2
Effectifs		33	67
Moyenne du nombre de pas		26350	40467
État des relations	Decision-achat	8,4	-4,04
	Application-prescription	-6,96	-0,31
	Investissement-dans-prod	9,77	9,67
	Contrôle-application-presc	-1,22	-1,55
	Nature-prescription	8,29	7,51
	Contrôle-nature-presc	0,73	-1,09
Satisfactions	CA	-11,88	57,76
	Père	57,8	57,08
	André	45,98	54,25
	Jean-BE	49,78	5,82
	Globale	141,68	174,91

Tableau 5.6. *Moyenne de l'état des relations et la satisfaction des acteurs dans les deux modes.*

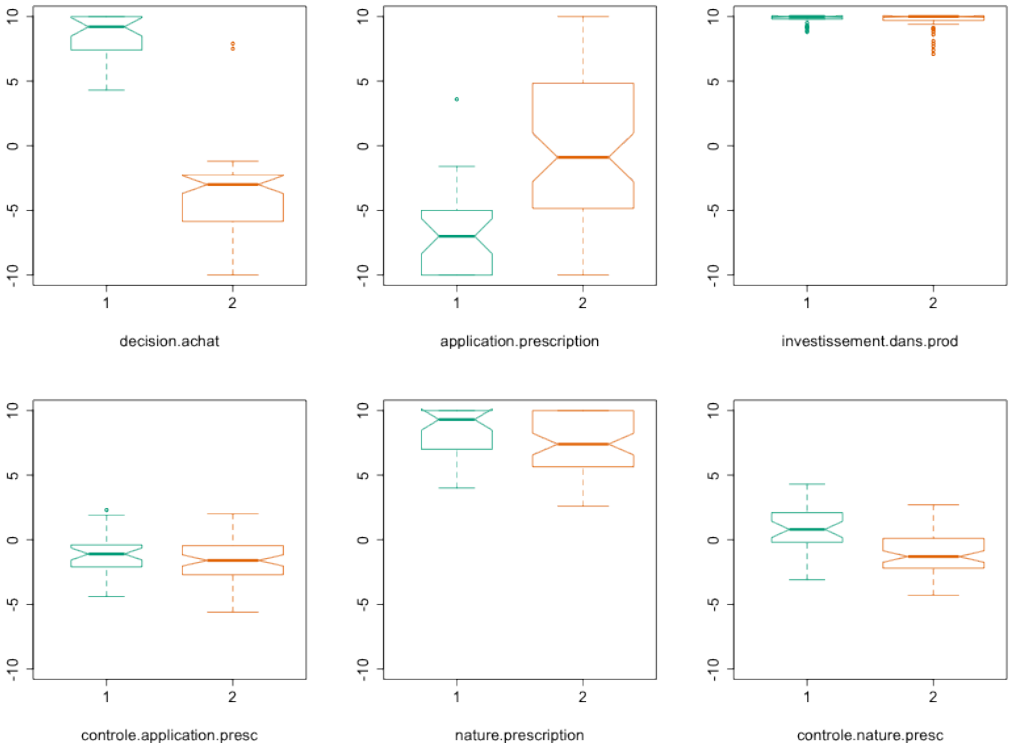


Figure 5.2. *Boxplot de l'état des relations dans le mode 1 (à gauche) et le mode 2 (à droite).*

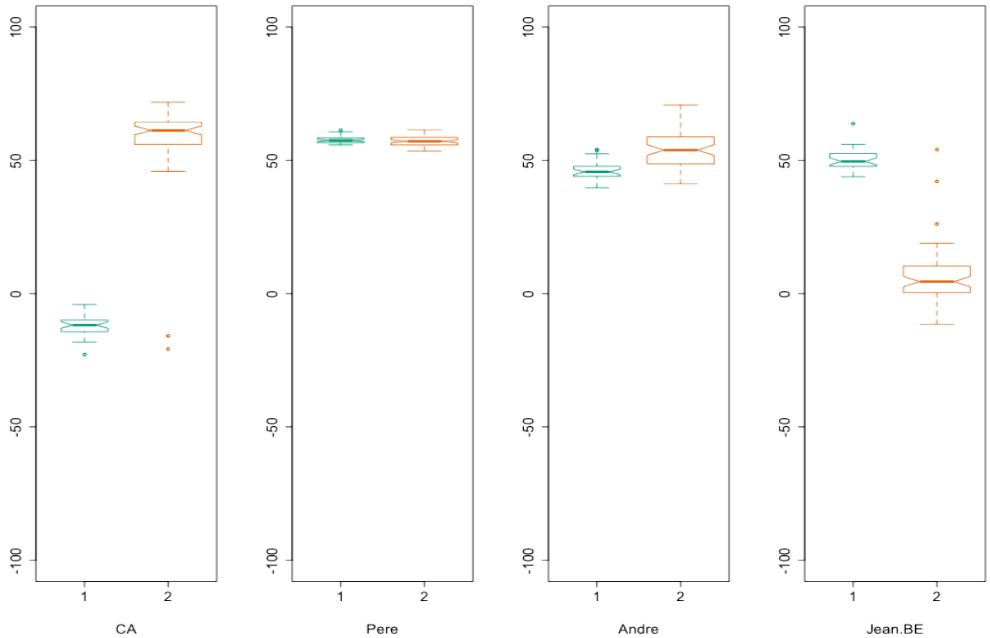


Figure 5.3. *Boxplot de la satisfaction des acteurs dans le mode 1 (à gauche) et le mode 2 (à droite).*

L'analyse de sensibilité de l'égocentrisme des acteurs (cf. 5.2.3 ci-dessous) montre que plus la valeur de ce paramètre diminue, plus les acteurs cherchent à maximiser la satisfaction globale de l'organisation, et plus les simulations convergent vers le deuxième mode.

Le modèle free-rider

Le cas traité dans cette section est le modèle free-rider où A1 utilise l'algorithme principal tandis que les autres acteurs utilisent l'algorithme de l'objectif global. Le tableau 5.7 présente les

résultats de 100 simulations du modèle free-rider avec cet algorithme, les résultats obtenus avec l'algorithme principal présentés dans 4.4.3 ainsi que, dans la dernière ligne, les satisfactions des quatre acteurs et la satisfaction globale dans l'état qui maximise la satisfaction globale. Les paramètres psycho-cognitifs et psycho-social sont initialisés de façon standard : discernement = 1, ténacité = réactivité = 5, égocentrisme = 0,5, et une répartition de 10% pour la dernière règle et 90% pour l'avant dernière règle. 100% des simulations ont convergé en moins de 60 000 pas.

		A1	A2	A3	A4	Satisfaction Globale
Algorithme de l'objectif global (A2, A3 et A4)	Satisfaction	51,2	82,6	82,6	84,4	300,8
	Satisfaction en %	75,6 %	91,3 %	91,3 %	92,2 %	97 %
	Ecart-Type	29,95	4,52	4,52	6,86	
Algorithme Principal pour les quatre acteurs	Satisfaction	25,1	85,9	86,3	86,1	283,4
	Satisfaction en %	62,55%	92,95%	93,15%	93,05%	94,28 %
	Ecart-Type	11,53	8,32	8,63	8,48	
Satisfaction Globale Maximale (Configuration C1)		80	80	80	80	**320**

Tableau 5.7. *Satisfaction des acteurs à l'issue de 100 simulations du modèle free-rider avec l'algorithme de l'objectif global, celles obtenues avec l'algorithme principal ainsi que, dans la dernière ligne, les satisfactions dans l'état qui maximise la satisfaction globale.*

Les résultats de simulations obtenus avec l'algorithme de l'objectif global favorise l'acteur A1 et défavorise légèrement les autres acteurs par rapport à ceux obtenus avec l'algorithme principal. La configuration C1, correspondant au maximum de la satisfaction globale, est atteinte dans 48% des simulations, alors qu'il ne l'est que dans 10,5% des simulations avec l'algorithme principal. A1 ne parvient pas toujours à imposer aux trois autres acteurs de coopérer systématiquement en sa faveur, seul comportement qui permet d'atteindre le maximum la satisfaction globale. Ayant un égocentrisme de 0,5, une telle situation n'est pas toujours acceptable par les autres acteurs qui, chacun à tour de rôle, bénéficient de la coopération des deux autres pour maximiser leur propre satisfaction.

L'analyse de sensibilité de l'égocentrisme des acteurs A2, A3 et A4 (cf. 5.2.3) montre que plus la valeur de ce paramètre diminue, plus ils cherchent à maximiser la satisfaction globale de l'organisation et coopèrent avec l'acteur A1, et plus la satisfaction globale se rapproche de son optimal.

5.2.3 – Analyse de sensibilité du paramètre égocentrisme

Le jeu considéré dans cette section est le modèle free-rider traité ci-dessus, où A1 utilise l'algorithme principal, tandis que les trois autres acteurs utilisent l'algorithme de l'objectif global. La figure 5.4 montre les résultats d'une analyse de sensibilité, comprenant 10 expériences, dont chacune a été réalisée avec 100 simulations, où l'égo-centrisme des trois acteurs (A2, A3 et A4) est le même et varie de 0,1 à 1. Les paramètres psycho-cognitifs sont initialisés de façon standard comme précédemment.

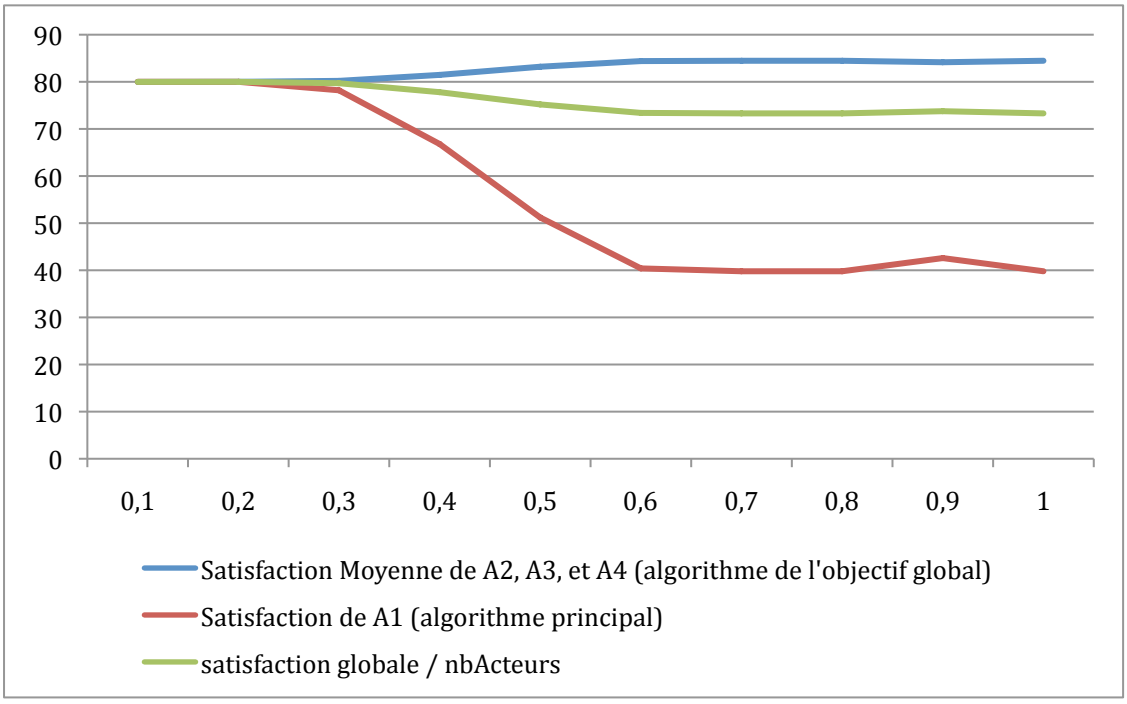


Figure 5.4. *La moyenne de la satisfaction de A1, qui utilise l'algorithme principal, de la moyenne des satisfactions des trois autres acteurs, qui utilisent l'algorithme de l'objectif global, et de la satisfaction globale, en fonction de l'égocentrisme des trois acteurs A2, A3, et A4.*

La figure 5.4 montre que plus l'égocentrisme de A2, A3 et A4 diminue, plus ils contribuent à l'augmentation de la satisfaction globale, et par conséquent plus la satisfaction de A1 augmente (sans pour autant faire baisser beaucoup leur satisfaction, du fait de la structure du jeu). En deçà de 0,3 d'égocentrisme, quasiment toutes les simulations convergent vers la configuration C1.

Nous avons aussi réalisé une analyse de sensibilité sur le cas Bolet où tous les acteurs utilisent l'algorithme de l'objectif global. La figure 5.5 montre les résultats de cette analyse, comprenant 10 expériences où l'égocentrisme est le même pour tous les acteurs et varie entre 0,1 et 1. Les paramètres psycho-cognitifs et psycho-social sont initialisés de façon standard comme précédemment.

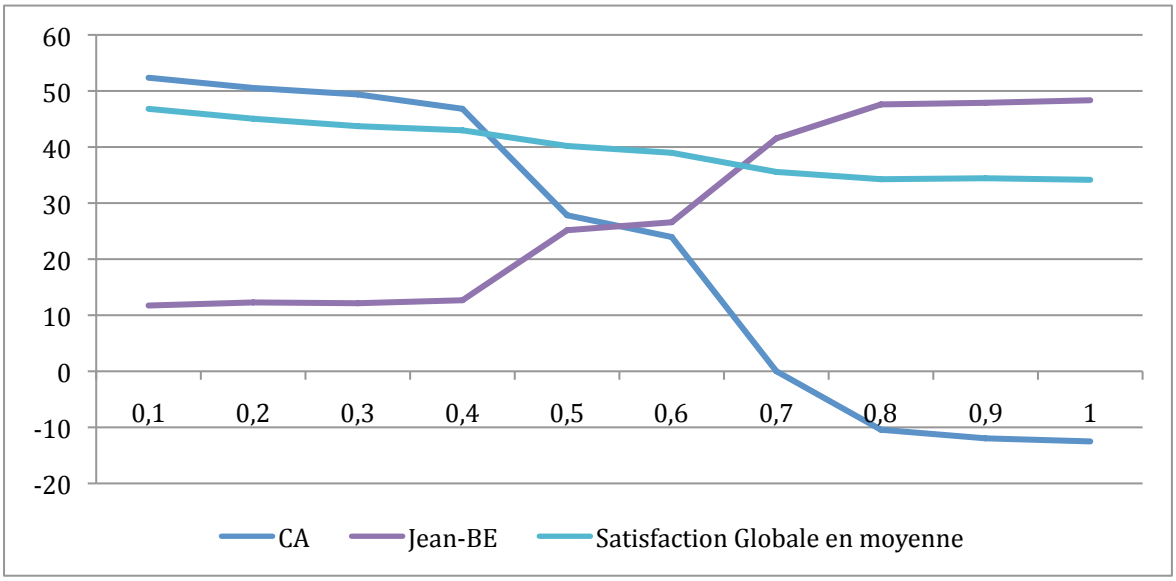


Figure 5.5. *La moyenne de la satisfaction de CA, Jean-BE, et de la moyenne de la satisfaction globale en fonction de l'égocentrisme des quatre acteurs.*

La figure 5.5 montre que plus l'égocentrisme des acteurs diminue, plus ils s'intéressent à l'augmentation de la satisfaction globale au lieu de leur propre satisfaction, plus les simulations convergent vers le deuxième mode de fonctionnement de cette organisation (cf. tableau 5.6), et par conséquent plus la satisfaction de CA augmente et celle de Jean-BE diminue, tandis que la satisfaction du père et d'André est quasiment constante.

L'augmentation de la satisfaction globale est due essentiellement au changement du comportement du père et du CA (cf. figure 5.6). En effet, ces deux acteurs contrôlent deux relations importantes (10 point d'enjeux sur la décision d'achat et 5,5 sur l'application de la prescription). L'effet de chacune des deux relations n'est pas le même sur les quatre acteurs, et l'état qui maximise l'impact de cette relation sur la satisfaction globale pénalise celui qui la contrôle. Plus l'égocentrisme du contrôleur diminue, plus il est prêt à sacrifier l'effet pour lui même de la relation qu'il contrôle en faveur de la satisfaction globale.

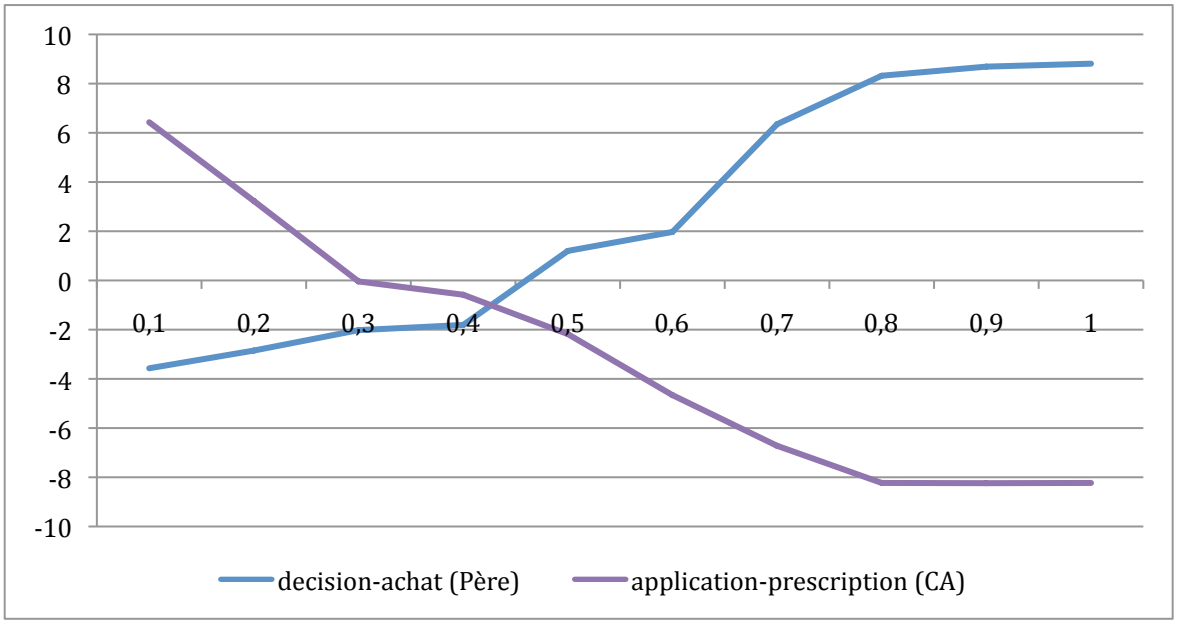


Figure 5.6. *La moyenne de l'état des deux relations (décision-achat et application-prescription) en fonction de l'égocentrisme des quatre acteurs.*

5.2.4 – Discussion

La variation de la satisfaction globale est calculée comme la somme de la variation des satisfactions de tous les acteurs :

$$\Delta satisfactionGlobale_t = \sum_{a \in A} \Delta satisfaction_t(a)$$

Cette forme d'agrégation de la satisfaction globale ne permet pas de savoir si la satisfaction de chaque acteur progresse autant qu'elle le pourrait. Les acteurs ne sont pas toujours en mesure de détecter l'optimalité de l'augmentation de la satisfaction globale, c'est-à-dire s'il existe une autre action conjointe qui conduirait à une augmentation plus grande que celle obtenue avec l'action conjointe appliquée.

Dans les deux cas présentés ci-dessus (le cas Bolet et le modèle free-rider), plus l'égocentrisme des acteurs diminue, plus ils ont tendance à maximiser la satisfaction globale même si cela les pénalise, et plus la satisfaction globale augmente. Cependant cette satisfaction n'atteint pas toujours son optimal (cf. figure 5.5) en raison de l'imprécision de l'évaluation de sa valeur et de la capacité cognitive des acteurs dans la réalisation de leur objectif ; toutes les analyses ont été faites avec les valeurs standards des paramètres psycho-cognitifs.

5.3 – L'élitiste et l'anti-élitiste

L'élitisme (respectivement anti-élitisme) est une attitude qui consiste à favoriser (respectivement défavoriser, pénaliser) les personnes occupant les meilleures positions, les élites. Nous considérons donc ici que l'élite est constituée de l'acteur ayant la satisfaction la plus élevée (on pourrait tout aussi bien considérer qu'il s'agit de l'acteur ayant la capacité d'action, le pouvoir ou l'influence la plus élevée).

Dans un système d'action concret, chaque acteur élitiste perçoit à chaque pas de la simulation la satisfaction de l'acteur le plus satisfait et participe, plus ou moins en fonction de son égocentrisme, à la réalisation de l'objectif de cet acteur, c'est-à-dire à la maximisation de sa satisfaction. De même, chaque acteur anti-élitiste cherche à empêcher l'acteur le plus satisfait de réaliser son objectif, c'est-à-dire tente de diminuer sa satisfaction.

5.3.1 – Présentation de l'algorithme

Cet algorithme, que l'on nomme *algorithme (anti-)élitiste*, repose sur l'utilisation par les acteurs élitistes ou anti-élitistes d'informations sur l'acteur le plus satisfait. En effet, un acteur élitiste « acteurE » ou anti-élitiste « acteurAE » doit être en mesure d'évaluer la situation de l'acteur le plus satisfait « acteurMax », afin de participer à l'amélioration (respectivement la détérioration) de la satisfaction de ce dernier. Pour cela, l'acteur le plus satisfait doit répondre à deux questions principales, et communiquer les deux variables correspondantes :

- Est-ce que ça va mieux qu'avant (Δ*satisfaction*) ?
- Es-tu dans une « bonne » situation (*écart*) ?

Ces deux variables permettent à l'acteur élitiste où anti-élitiste d'intégrer l'évaluation de la situation de l'élite dans celle de sa situation courante de la façon suivante :

- Un élitiste calcule la variation de sa satisfaction en fonction de la variation de sa satisfaction personnelle, ΔsatisPerso (cf. éq. 4.17), et de celle de la satisfaction de l'acteurMax de la façon suivante :

$$\Delta satisfaction_t\,(acteurE) = EC \times \Delta satisPerso_t\,(acteurE) + (1 - EC) \times \Delta satisfaction_t\,(acteurMax) \qquad \text{(éq. 5.13)}$$

Un anti-élitiste qui n'est pas l'acteur le plus satisfait s'intéresse à diminuer la satisfaction de l'acteurMax ; il calcule donc la variation de sa satisfaction en fonction de la variation de sa propre satisfaction de laquelle il soustrait la variation de la satisfaction de l'acteurMax de la façon suivante :

$$\Delta satisfaction_t\,(acteurAE) = EC \times \Delta satisPerso_t\,(acteurAE) - (1 - EC) \times \Delta satisfaction_t\,(acteurMax) \qquad \text{(éq. 5.14)}$$

S'il est l'acteur le plus satisfait, il ne cherche pas à diminuer sa propre satisfaction, et donc[22] :

$$\Delta satisfaction_t\,(acteurAE) = \Delta satisPerso_t\,(acteurAE) \qquad \text{(éq. 5.15)}$$

- Un élitiste calcule son écart à chaque instant en fonction de son écart personnel (cf. éq. 4.4) et de celui de l'écart de l'acteurMax :

$$écart_t\,(acteurE) = EC \times écartPerso_t\,(acteurE) + (1 - EC) \times écart_t\,(acteurMax) \qquad \text{(éq. 5.16)}$$

[22] Bien évidemment, rien ne s'oppose à qu'un acteur anti-élitiste n'applique sa rationalité même lorsqu'il est en position d'élite, l'équation 4.14 devenant alors $\Delta satisfaction_t\,(acteurAE) = -\,\Delta satisPerso_t\,(acteurAE)$.

Inversement, un anti-élitiste qui n'est pas l'acteur le plus satisfait calcule son écart à chaque instant en fonction de son écart personnel et de l'opposé de celui de l'acteurMax :

$$\text{écart}_t\,(\text{acteurAE}) = EC \times \text{écartPerso}_t\,(\text{acteurAE}) + (1 - EC) \times (1 - \text{écart}_t\,(\text{acteurMax})) \qquad \text{(éq. 5.17)}$$

S'il est l'acteur le plus satisfait, il applique la formule suivante :

$$\text{écart}_t\,(\text{acteurAE}) = \text{écartPerso}_t\,(\text{acteurAE}) \qquad \text{(éq. 5.18)}$$

- Les autres variables de cet algorithme (l'ambition, le taux d'exploration, l'intensité des actions, et les forces des règles) sont calculées et mises à jour de la même façon que dans l'algorithme principal.

Un élitiste qui se trouve être l'acteur ayant la satisfaction maximale est satisfait si sa propre satisfaction est supérieure ou égale à son ambition ; dans le cas contraire il n'est satisfait que si l'acteurMax est satisfait ($\text{écart}_t(\text{acteurMax}) \leq 0$). Par contre un anti-élitiste est satisfait si sa propre satisfaction est plus grande que son ambition. Cette dissymétrie entre les critères de satisfaction d'un acteur élitiste et d'un acteur anti-élitiste provient de ce que, pour ce dernier, il n'est pas possible de concilier en une seule grandeur la recherche d'une maximisation (sa satisfaction personnel) et celle d'une minimisation (la satisfaction de l'acteurMax, pour laquelle on ne dispose pas de plancher raisonnable, ce qui fait que l'on ne peut pas définir de seuil).

Notons que notre version de la rationalité des acteurs anti-élitistes est s'apparente à la jalousie ; un acteur anti-élitiste change son attitude en fonction de l'identité de l'acteur le plus satisfait : s'il est le plus satisfait, il cherche à maximiser sa propre satisfaction. Tandis que, dans le cas contraire, il tente de détériorer la satisfaction de l'acteurMax même si cela est pénalisant pour lui. En effet, le fait qu'un acteur diminue sa propre satisfaction de façon intentionnelle et sans participation à la réalisation d'un objectif globale ne nous semble pas plausible sociologiquement. Dans certains cas, il se peut qu'un acteur applique des actions considérées comme bonne pour son objectif global, mais qui font diminuer inconsciemment sa propre satisfaction. Pour résumer, un acteur anti-élitiste s'intéresse à être lui-même le plus satisfait.

Cet algorithme ainsi que les deux algorithmes suivantes diffèrent de l'algorithme principal dans les étapes 1 et 4. Les étapes de l'algorithme sont les suivantes :

Initiation :

L'état de chaque relation est initialisé arbitrairement à 0 (l'état neutre).

La satisfaction est calculée en fonction des états des relations dont l'acteur dépend (éq. 4.3)

L'ambition est initialisée à la valeur maximale de la satisfaction de l'acteur (éq. 4.6)

L'écart est calculé en fonction de la satisfaction et l'ambition (éq. 4.4 et 4.5)

Le taux d'exploration est initialisé à la valeur du taux d'exploration instantané (éq. 4.9, 4.10, et 4.11)

A chaque étape t de la simulation, chaque acteur :

1. perçoit sa satisfaction (éq. 4.3), calcule son écart (éq. 4.4, 5.16, 5.17 et 5.18), et met à jour son ambition (éq. 4.7 et 4.8).

2. met à jour son taux d'exploration (éq. 4.9, 4.10, 4.11, et 4.12).

3. met à jour l'intensité des actions (équation 4.13).

4. met à jour la force des deux dernières règles appliquées (éq. 4.15, 4.16, 4.17, 5.13, 5.14 et 5.15). Les règles de force négative sont oubliées.

5. sélectionne les règles applicables, celles dont la composante situation est proche de sa situation courante en fonction de son discernement (4.2).

6. Si l'ensemble des règles applicables est vide (notamment au début de la simulation), il crée une nouvelle règle, avec une force initialisée à 0, une situation égale à la situation courante et les actions (modifications à apporter sur les états des relations contrôlées) choisies au hasard dans l'intervalle [- intensité ; + intensité] (éq. 4.14).

7. choisit la règle nouvellement créée ou, parmi celles applicables, l'une de trois règles dont la force est la plus grande.

Lorsque tous les acteurs ont choisi une règle, leurs actions sont appliquées.

Algorithme 5.3. *Schéma de l'algorithme de rationalité des acteurs élitistes et anti-élitistes.*

Cet algorithme suppose qu'un acteur élitiste ou anti-élitiste est capable d'identifier l'acteur ayant la satisfaction maximale ou minimale d'une part, et d'autre part d'évaluer la progression de cet acteur dans la réalisation de son objectif (ΔSatisfaction) et l'écart entre sa satisfaction et son ambition. La perception de l'acteurMax et de sa ΔSatisfaction correspond à une quantité d'information sur l'état du jeu qui est du même ordre que celle qui permet à un acteur d'exercer des solidarités. En outre, le surcroît d'information sur l'écart d'un autre acteur semble réaliste.

5.3.2 – Exemples d'application / élitiste

Afin d'évaluer la qualité de cet algorithme, nous analysons son comportement sur le dilemme du prisonnier et le modèle free-rider.

Le dilemme du prisonnier

Le dilemme du prisonnier traité dans cette analyse est celui avec une répartition 1/9 où les deux acteurs sont élitistes. On s'attend donc à ce que les deux acteurs coordonnent leurs actions afin de maximiser la satisfaction de l'élite quelque qu'il soit.

Le tableau 5.8 présente les résultats de 200 simulations de ce modèle avec cet algorithme. Les paramètres psycho-cognitifs et psycho-social sont initialisés de façon standard : discernement = 1, ténacité = réactivité = 5, égocentrisme = 0,5, et une répartition de 10% pour la dernière règle et 90%

pour l'avant dernière règle. Toutes les simulations ont convergé en moins de 2 000 pas avec une moyenne de 687 pas.

		A1	A2	acteurMax	autre acteur
satisfaction	Moyenne	32,63	34,33	**91,46**	-24,49
	Maximum	100	100	100	80
	Minimum	-100	-100	80	-100
	Ecart-type	74,59	75,04	8,82	80,61

Tableau 5.8. *Satisfaction des deux acteurs, de l'acteurMax et l'autre acteur à l'issue de 200 simulations du dilemme du prisonnier où les deux acteurs sont élitistes avec un égocentrisme de 0,5.*

Remarquons que la satisfaction de chaque acteur a atteint son maximum (100) et son minimum (-100). La satisfaction moyenne des acteurs est très faible par rapport à celle obtenue avec l'algorithme principal (cf. tableau 4.3). En effet, dans notre façon de modéliser les organisations, le fait de favoriser l'un des acteurs ne crée pas nécessairement de capacité d'action supplémentaire, et l'augmentation de la satisfaction de l'acteur en position d'élite se fait alors au détriment d'un ou plusieurs autres acteurs.

En ce qui concerne la satisfaction de l'acteurMax dans chaque simulation, elle varie entre 80 et 100 avec une moyenne de 91,46, tandis que la satisfaction de l'autre acteur (ou bien le min (A1, A2)) varie entre -100 et 80 avec une moyenne de -24,49. En effet, chacun des acteurs, ayant un égocentrisme de 0,5, s'intéresse à maximiser la satisfaction de l'acteur le plus satisfait autant qu'à augmenter la sienne. 36% des simulations convergent vers un optimum de Pareto (une bonne satisfaction pour les deux acteurs égale à 80), tandis que les autres simulations (64%) ont convergé vers une configuration élitiste avec une satisfaction d'un acteur parmi les deux proche de son optimal (plus grand que 85) tandis que la satisfaction de l'autre acteur est très faible et parfois proche de son pire.

Une Classification Ascendante Hiérarchique permet de bien caractériser les trois modes vers lesquels les simulations convergent. Le tableau 5.9 montre la moyenne de l'état des relations et de la satisfaction des acteurs pour chacun des modes, et les figures 5.7 et 5.8 les boxplots des relations et des acteurs dans chacun des trois modes. Il apparaît clairement que c'est l'acteur A1 en position d'élite dans le premier mode, l'acteur A2 le plus satisfait dans le deuxième mode, tandis que, dans le troisième mode, les simulations ont convergé vers une configuration non élitiste qui procure une bonne satisfaction à chacun des deux acteurs, mais après un nombre de pas beaucoup plus important.

		Mode 1	Mode 2	Mode 3
Effectifs		56	57	87
Moyenne du nombre de pas		71	125	1451
État des relations	R1	-9,66	9,78	9,49
	R2	9,79	-9,53	9,17
Satisfactions	A1	99,5	-94,91	73,15
	A2	-96,79	99,06	76,33

Tableau 5.9. *Moyenne de l'état des relations et la satisfaction des acteurs dans les trois modes.*

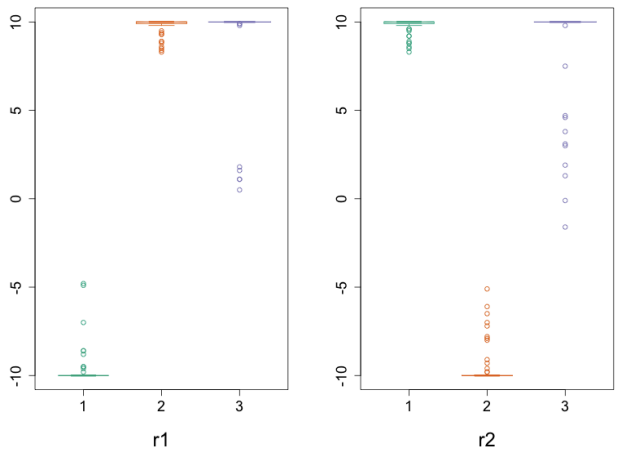


Figure 5.7. *Boxplot de l'état des relations dans le premier mode (à gauche), le deuxième mode (au centre) et le troisième mode (à droite).*

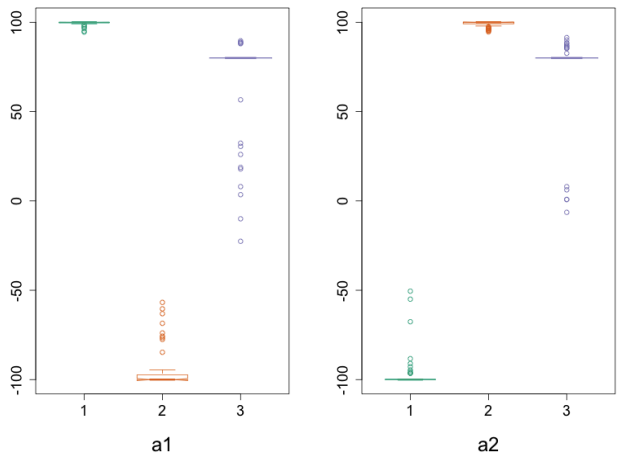


Figure 5.8. *Boxplot de la satisfaction des acteurs dans le premier mode (à gauche), le deuxième mode (au centre) et le troisième mode (à droite).*

L'analyse de sensibilité dans 5.3.4 ci-dessous montre que plus l'égocentrisme des deux acteurs diminue (plus ils sont élitistes), plus les simulations convergent vers les deux premiers modes, et plus la moyenne du maximum de la satisfaction des deux acteurs (la satisfaction de l'acteurMax) augmente.

Le modèle free-rider

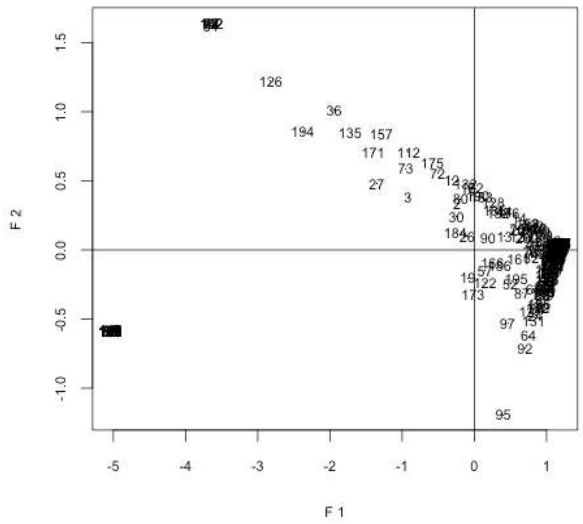
Le cas traité dans cette section est le modèle free-rider où A1 utilise l'algorithme principal, tandis que les trois autres acteurs sont élitistes avec un égocentrisme de 0,5. On s'attend donc à ce que l'acteur A1 profite de la coopération envers lui des trois autres acteurs, et arrive dans la plupart des cas à maximiser sa propre satisfaction.

Le tableau 5.10 présente les résultats de 200 simulations de ce modèle. Les paramètres psycho-cognitifs sont initialisés de façon standard comme précédemment. Toutes les simulations ont convergé en moins de 6 000 pas avec une moyenne d'environ 980 pas. Les configurations atteintes sont bien élitistes, puisque la moyenne de la satisfaction de l'acteur le plus satisfait est de 94,31 avec un faible écart-type.

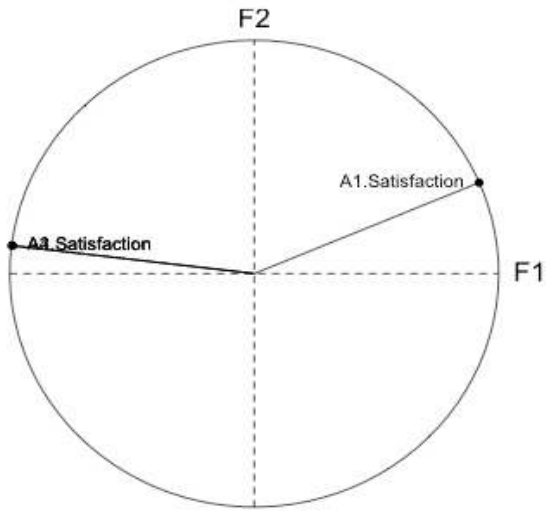
		A1	A2	A3	A4	acteurMax
satisfaction	Moyenne	86,31	-63,41	-63,44	-63,65	94,31
	Maximum	100	100	100	100	100
	Minimum	20	-100	-100	-100	66,7
	Ecart-type	14,44	46,78	46,44	46,5	4,85

Tableau 5.10. *Satisfaction des acteurs à l'issue de 200 simulations du modèle free-rider où A1 utilise l'algorithme principal, tandis que les autres acteurs sont élitistes avec un égocentrisme de 0,5.*

La figure 5.9 et le tableau 5.11 montrent une analyse en composantes principales des résultats de simulations présentés dans le tableau 5.10.



(a) représentation graphique des résultats de simulations (en ne considérant que la satisfaction des acteurs) dans un repère formé des deux axes associés aux deux premières composantes principales (F1 et F2)



(b) représentation graphique des variables (les satisfactions des quatre acteurs ; celles des trois acteurs A2, A3 et A4 sont juxtaposées à gauche) par rapport au cercle unité dans un repère dont les deux axes sont les deux premières composantes principales (F1 et F2)

Figure 5.9. *Analyse en composantes principales des résultats de simulation du modèle free-rider obtenue avec le logiciel R (99,82 de la variance est expliquée).*

Le tableau 5.11 montre que le premier Axe principal est une opposition entre la satisfaction de A1 et celle de trois autres acteurs (A2, A3 et A4). Tandis que le deuxième axe n'est pas significatif, puisque 95% de la variance est expliquée par le premier axe.

	Satisfaction			
	A1	A2	A3	A4
F1	0,92	-0,99	-0,99	-0,99
F2	0,39	0,12	0,12	0,12

Tableau 5.11. *Contribution des variables (les satisfactions des acteurs) aux deux composantes principales F1 et F2.*

L'acteur A1 utilise l'algorithme principal, et ne s'intéresse qu'à augmenter de sa propre satisfaction, tandis que les autres acteurs sont élitistes avec un égo-centrisme de 0,5 ; ils s'intéressent donc à l'augmentation de la satisfaction de l'acteur le plus satisfait, A1 dans la plupart des cas, autant qu'à l'augmentation de leur propre satisfaction. Dans 12% des simulations, un des trois acteurs (A2, A3 ou A4) parvient à maximiser sa propre satisfaction (cf. figure 5.9.a, l'agglomérat de points sur la gauche en bas). Par contre, il n'arrive jamais que les trois acteurs, ou deux parmi les trois, parviennent à maximiser leur propre satisfaction en même temps, ce qui serait inacceptable par A1. Dans les autres simulations (88%), A1 profite de la participation inconditionnelle d'au moins deux des autres acteurs à l'augmentation de sa satisfaction, et préserve plus ou moins, dans 70% des simulations, l'impact pour lui-même de la relation qu'il contrôle. Notons que parmi les 88%, 3% des simulations (l'agglomérat de points sur la gauche en haut de la figure 5.9.a) convergent vers un état proche du maximum global (configuration C1).

Nous avons aussi considéré le cas où tous les acteurs sont élitistes avec un égocentrisme de 0,5. Le tableau 5.12 présente les résultats de 200 simulations du modèle free-rider avec cet algorithme. Les paramètres psycho-cognitifs sont initialisés de façon standard comme précédemment. Toutes les simulations ont convergé en moins de 2 000 pas avec une moyenne de 240 pas.

		A1	A2	A3	A4	acteurMax
satisfaction	Moyenne	10,23	47,9	46,39	47,57	**97,3**
	Maximum	100	100	100	100	100
	Minimum	-100	-100	-100	-100	80
	Ecart-type	53,41	66,98	66,26	66,8	3,41

Tableau 5.12. *Satisfaction des quatre acteurs, de l'acteurMax et l'acteurMin à l'issue de 200 simulations du modèle free-rider où tous les acteurs sont élitistes avec un égocentrisme de 0,5.*

Les configurations atteintes sont encore plus élitistes, avec 97,3 de satisfaction en moyenne pour l'acteurMax. Chacun des acteurs est le plus satisfait dans un certain nombre de simulations. Les satisfactions des trois acteurs (A2, A3 et A4) sont quasiment les mêmes, et nettement supérieures à la satisfaction de A1. En effet, l'état qui maximise la satisfaction de A1 place les autres acteurs dans leur pire situation, tandis que l'état qui maximise la satisfaction d'un parmi les trois acteurs (A2, A3 et A4) est bénéfique aux deux autres, ce qui conduit à une satisfaction moyenne des trois acteurs supérieure à celle de A1.

Une Classification Ascendante Hiérarchique permet de caractériser deux modes vers lesquels les simulations convergent. Le tableau 5.13 montre la moyenne de l'état des relations et de la satisfaction des acteurs pour chacun des modes, et les figures 5.10 et 5.11 les boxplots des relations et des acteurs dans chacun des modes. Il apparaît clairement que l'acteur A1 est en position d'élite dans le premier mode, tandis que dans le deuxième mode un des trois acteurs est en position d'élite et les deux autres ont une « bonne » satisfaction, légèrement inférieure à celle de l'acteur en position d'élite, grâce à la coopération de A1.

		Mode 1	Mode 2
Effectifs		23	77
Nombre de pas		170	261
État des relations	R1	-9,72	9,89
	R2	10	-1,6
	R3	9,93	0,32
	R4	9,93	-1,06
Satisfactions	A1	99,2	-16,35
	A2	-97,71	91,39
	A3	-97,64	89,42
	A4	-97,65	90,95

Tableau 5.13. *Moyenne de l'état des relations et la satisfaction des acteurs dans les deux modes.*

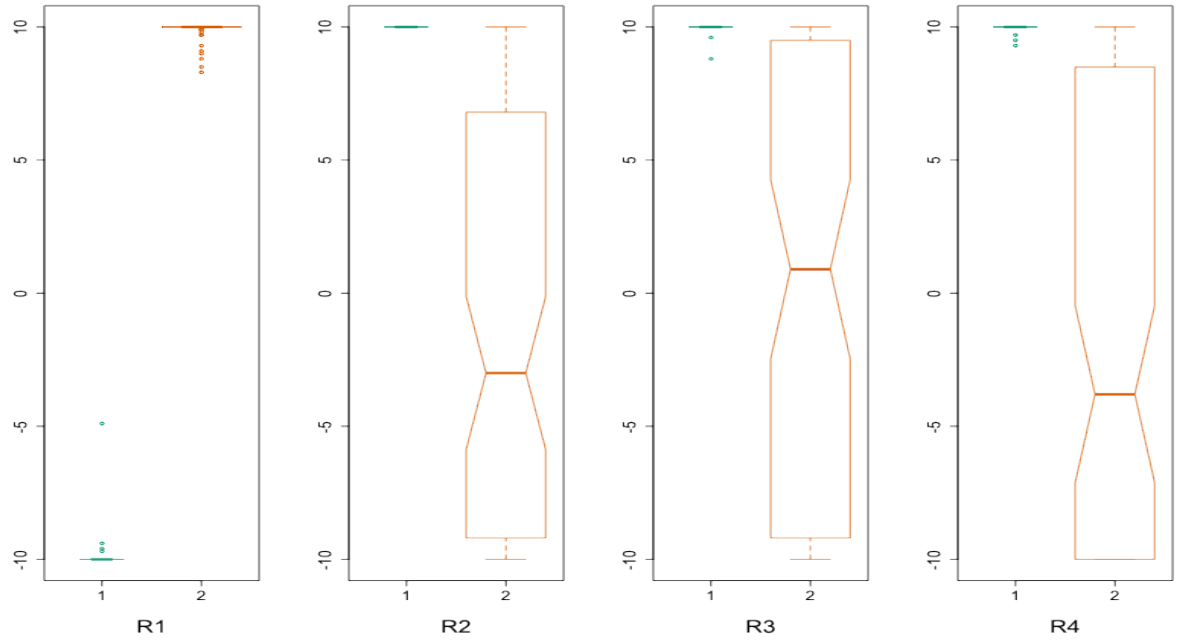


Figure 5.10. *Boxplot de l'état des relations dans chacun des deux modes.*

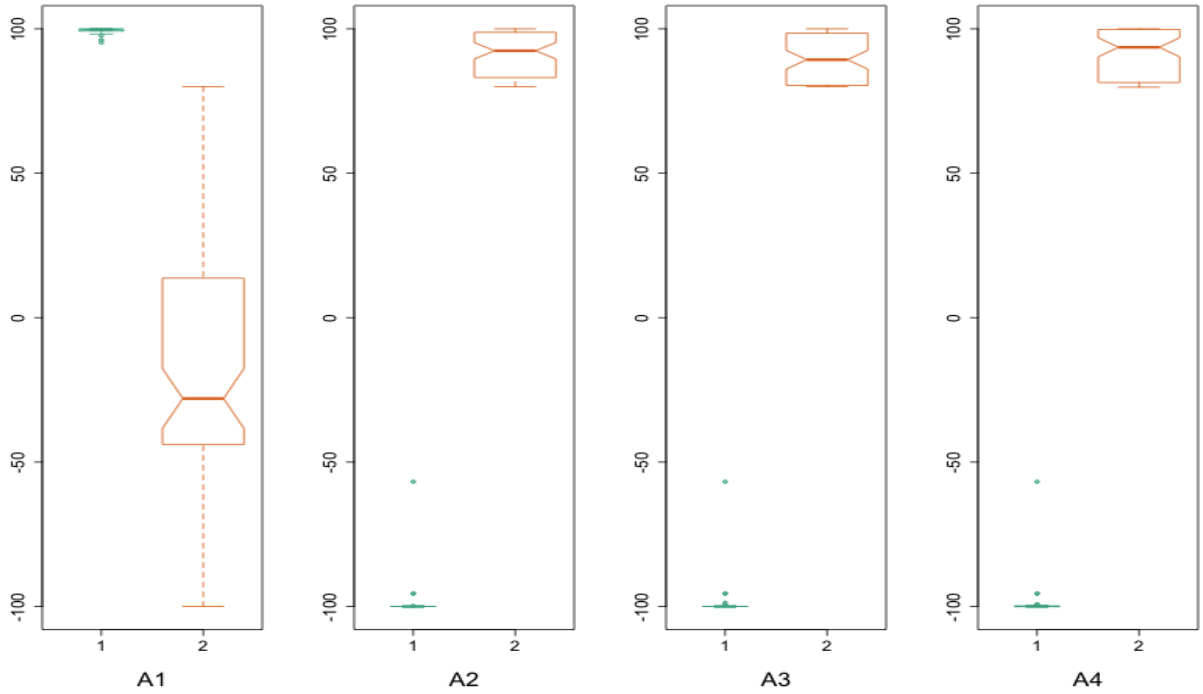


Figure 5.11. *Boxplot de la satisfaction des acteurs dans chacun des deux modes.*

Il apparaît que, dans le cas où seuls les acteurs A2, A3 et A4 sont élitistes, la convergence des simulations tient à ce que A1 d'une part contrôle la relation la plus pertinente de l'organisation de laquelle les trois acteurs dépendent et d'autre part utilise l'algorithme principal, et cherche toujours à maximiser sa propre satisfaction sans se préoccuper de celles des autres. A1 est donc capable de se placer en position d'élite de l'organisation, ce qui suffisant et nécessaire pour que les autres coopèrent avec lui.

5.3.3 – Exemples d'application / anti-élitiste

Nous analysons le comportement de cet algorithme sur les mêmes modèles : le dilemme du prisonnier et le modèle free-rider.

Le dilemme du prisonnier

Le dilemme du prisonnier traité dans cette analyse est celui avec une répartition 1/9 où les deux acteurs sont anti-élitistes. On s'attend donc à ce que les acteurs luttent l'un contre l'autre sur la position d'élite, chacun cherchant à être le plus satisfait.

Le tableau 5.14 présente les résultats de 200 simulations de ce modèle avec cet algorithme et les résultats obtenus avec l'algorithme principal présentés dans 4.4.1. Les paramètres psycho-cognitifs et psycho-social sont initialisés de façon standard comme précédemment. 71% des simulations ont convergé en moins de 300 000 pas avec une durée moyenne de 1 600 pas.

		A1	A2	acteurMax
Algorithme anti-élitiste	Satisfaction	75,48	75,54	75,54
	Ecart-Type	8,77	8,66	8,66
Algorithme Principal	Satisfaction	80	80	80
	Ecart-Type	0	0	0

Tableau 5.14. *Satisfaction des acteurs à l'issue de 200 simulations du dilemme du prisonnier où les deux acteurs sont anti-élitistes avec un égocentrisme de 0,5 ainsi que celles obtenues avec l'algorithme principal.*

Un acteur anti-élitiste qui n'est pas l'acteur le plus satisfait cherche à diminuer la satisfaction de ce dernier, tandis que dans le cas contraire il cherche à améliorer sa propre satisfaction. Le jeu est donc à une compétition entre les deux acteurs où chacun cherche à être plus satisfait que l'autre. Dans 71% des simulations, les acteurs acceptent une régulation dans un état proche d'un optimum de Pareto et se contentent d'avoir une « bonne » satisfaction sans plus chercher à diminuer celle de l'autre. Dans les autres simulations (les plus longues), cette régulation est refusée par au moins un des deux acteurs. Les satisfactions des deux acteurs sont bloquées dans un intervalle proche de la satisfaction maximale ; elles alternent comme illustré dans la figure 5.12 et les simulations se prolongent au delà de 300 000 pas (environ 187 fois la moyenne de la durée des simulations qui ont convergé).

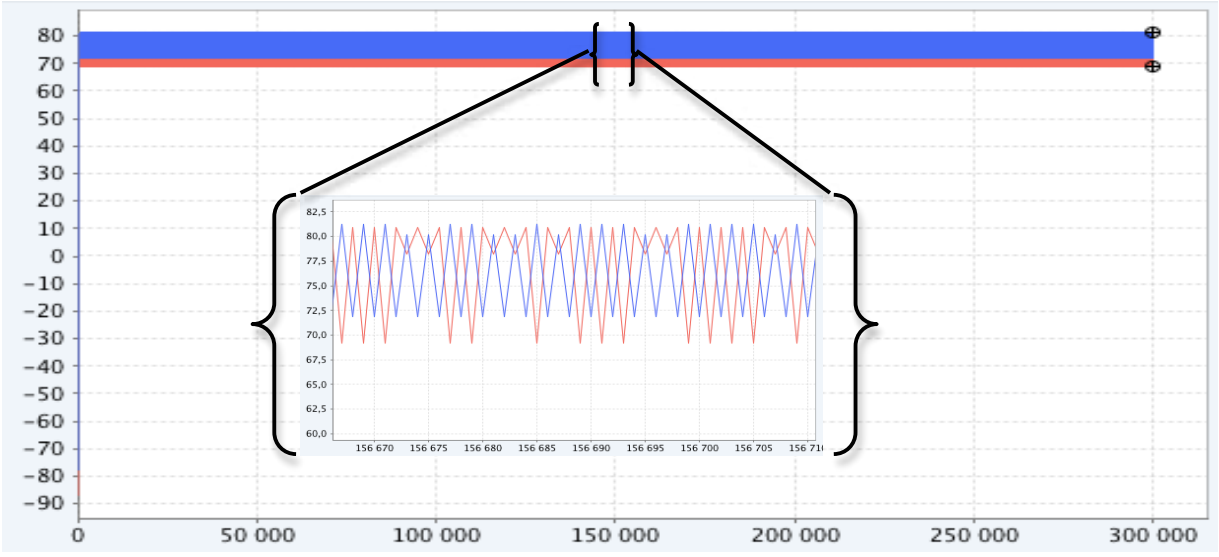


Figure 5.12. *La satisfaction des deux acteurs au cours de la 45ème simulation.*

Il apparaît que, dans ce modèle, l'anti-élitisme rallonge considérablement les simulations, sans diminuer sensiblement la satisfaction des acteurs. En effet, notre façon de définir l'anti-élitisme en fait des acteurs jaloux, rationalité qui ne s'accommode d'aucun compromis dans un jeu symétrique à deux acteurs. Ayant un égo-centrisme de 0,5, il se peut qu'un acteur anti-élitiste accepte un état qui lui procure une « bonne » satisfaction bien qu'il ne soit pas l'acteur le plus satisfait s'il perçoit que la compétition le conduit dans sa situation la pire.

Le modèle free-rider

Le cas traité dans cette analyse est le modèle free-rider où A1 utilise l'algorithme principal, tandis que les autres acteurs sont anti-élitistes avec un égocentrisme de 0,5. On s'attend à ce que les trois acteurs A2, A3 et A4 luttent entre eux sur la position d'élite, tandis que A1 profitera de cette compétition entre les trois acteurs afin d'augmenter sa propre satisfaction.

Le tableau 5.15 présente les résultats de 200 simulations de ce modèle ainsi que les résultats obtenus avec l'algorithme principal présentés dans 4.4.3. Les paramètres psycho-cognitifs sont initialisés de façon standard comme précédemment. Toutes les simulations ont convergé en moins de 10 000 pas avec une moyenne d'environ 2 710 pas.

		A1	A2	A3	A4	acteurMax	Satisfaction Globale
Algorithme anti-élitiste pour A2, A3 et A4	Satisfaction	78,89	80,02	80,01	80,05	80,22	318,97
	Satis. en %	89,45%	90,01%	90,01%	90,03%	90,11%	99,84%
	Ecart-Type	1,2	0,21	0,19	0,19	0,28	
Algorithme Principal (les quatre acteurs)	Satisfaction	25,1	85,9	86,3	86,1	97,9	283,4
	Satis. en %	62,55%	92,95%	93,15%	93,05%	98,95%	94,28%
	Ecart-Type	11,53	8,32	8,63	8,48	3,76	

Tableau 5.15. *Satisfaction moyennes des quatre acteurs et celle de l'acteurMax à l'issue de 200 simulations du modèle free-rider (A1 utilise l'algorithme principal, tandis que les autres acteurs sont anti-élitistes avec un égocentrisme de 0,5) ainsi que celles obtenues avec l'algorithme principal pour les quatre acteurs.*

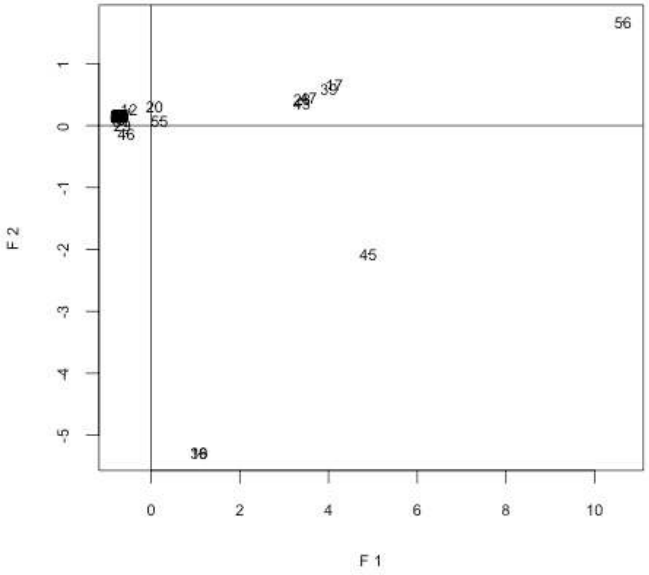
Ces résultats convergent vers la configuration correspondant au maximum de la satisfaction globale, et ce beaucoup mieux que l'algorithme de l'objectif global présenté en 5.2. De plus, ces simulations convergent assez rapidement. La structure du jeu fait qu'aucun des acteurs A2, A3 et A4 n'a une influence directe sur les deux autres, leur action n'affecte donc que la satisfaction de A1. Ils s'aperçoivent rapidement que la diminution de la satisfaction de l'acteurMax est très pénalisante pour eux-mêmes puisque A1 réagit en ne coopérant plus, et se contentent d'obtenir une bonne satisfaction. L'acteur A1 profite ainsi de la coopération des trois autres acteurs, et améliore largement sa satisfaction par rapport à celle obtenue dans (4.4.3) où les quatre acteurs utilisent l'algorithme principal. L'anti-élitisme empêche donc l'absence de coopération de l'un des trois acteurs que l'on observe dans 90% des cas avec l'algorithme principal, et ce au bénéfice de la satisfaction globale de l'organisation.

Nous avons aussi considéré le cas où tous les acteurs sont anti-élitistes avec un égocentrisme de 0,5. Le tableau 5.16 présente les résultats de 100 simulations du modèle free-rider avec cet algorithme avec les mêmes valeurs des paramètres psycho-cognitifs. 63% des simulations ont convergé en moins de 300 000 pas avec une moyenne d'environ 5 700 pas.

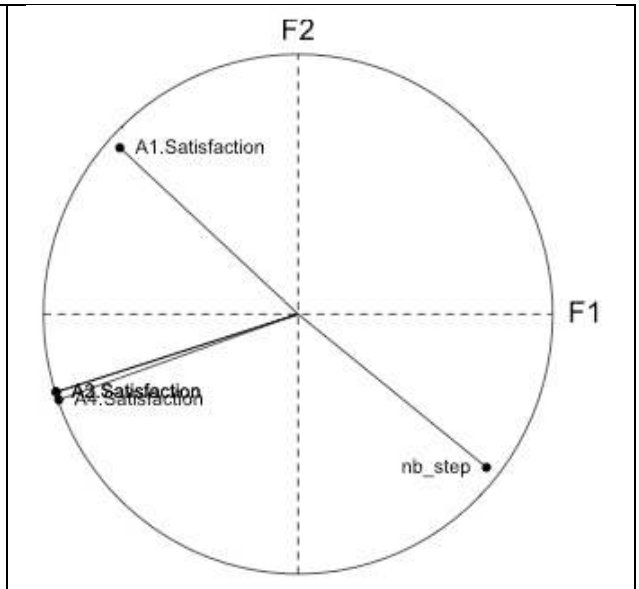
	A1	A2	A3	A4	acteurMax
Satisfaction	66,13	72,68	72,39	72,08	73,72
Ecart-type	22,45	13,7	14,24	14,77	12,15

Tableau 5.16. *Satisfaction des acteurs à l'issue de 200 simulations du modèle free-rider où tous les acteurs sont anti-élitistes avec un égocentrisme de 0,5.*

La figure 5.13 et le tableau 5.17 montrent une analyse en composantes principales des résultats de simulations présentés dans le tableau 5.16.



(a) représentation graphique des résultats de simulations (en considérant la satisfaction des acteurs et le nombre de pas) dans un repère formé des deux axes associés aux deux premières composantes principales (F1 et F2)



(b) représentation graphique des variables (les satisfactions des quatre acteurs et le nombre de pas) par rapport au cercle unité dans un repère dont les deux axes sont les deux premières composantes principales (F1 et F2)

Figure 5.13. *Analyse en composantes principales des résultats de simulation du modèle free-rider obtenue avec le logiciel R (95,22 de la variance est expliquée).*

Le tableau 5.17 montre que le premier Axe principal (74,31% d'explication) est une opposition entre le nombre de pas et la satisfaction des quatre acteurs. Tandis que le deuxième axe (20,91% d'explication) est une faible opposition entre la satisfaction de A1 d'une part, et la satisfaction des autres acteurs d'autre part.

	Nb_step	A1	A2	A3	A4
F1	0,74	-0,7	-0,95	-0,95	-0,94
F2	-0,59	0,64	-0,3	-0,3	-0,33

Tableau 5.17. *Contribution des variables (les satisfactions des acteurs) aux deux composantes principales F1 et F2.*

Parmi les 63% des simulations qui ont convergé, 55 (les plus courtes) ont convergé vers un état proche de la configuration C1, tandis que les 8 autres simulations (les plus longues) ont convergé vers un état qui procure une faible (parfois la pire) satisfactions aux quatre acteurs. Dans les 37% simulations qui n'ont pas convergé, chaque acteur s'acharne à détériorer la situation des

autres, comme s'il fallait que les simulations durent plus longtemps afin que la concurrence s'intensifie et aucun n'accepte plus un état dans lequel il n'est pas l'acteur le plus satisfait.

Il apparaît que, dans le cas où seuls les acteurs A2, A3 et A4 sont anti-élitistes, la convergence des simulations tient à ce que d'une part A1 n'est pas anti-élitiste et d'autre part il n'est pas en position d'élite. De par la structure du modèle, les trois acteurs n'ont aucune influence direct l'un sur les autres, leurs actions n'affectent directement que la satisfaction de A1 qui contrôle la relation dont les trois acteurs dépendent. Ils s'aperçoivent rapidement que la non coopération avec l'acteur A1 leur coûte très cher et, ayant un égocentrisme de 0,5, ils se trouvent obliger de coopérer avec A1.

5.3.4 – Analyse de sensibilité du paramètre égocentrisme / élitiste

Le jeu considéré dans cette section est le dilemme du prisonnier avec une répartition 1/9 où chacun des acteurs étant élitiste. La figure 5.14 montre les résultats d'une analyse de sensibilité, comprenant 10 expériences où le paramètre égocentrisme est le même pour les deux acteurs et varie entre 0,1 et 1. Les paramètres psycho-cognitifs sont initialisés de façon standard comme précédemment.

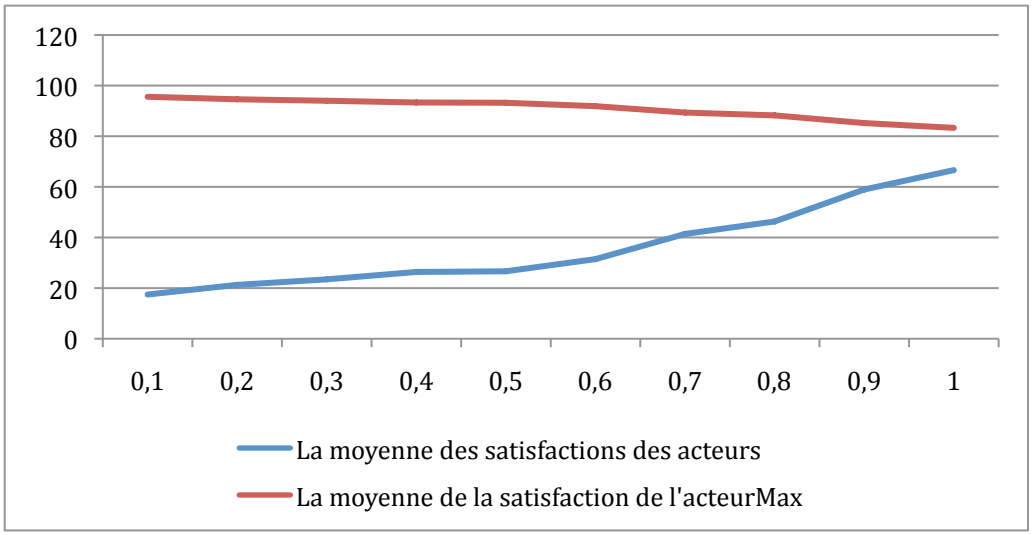


Figure 5.14. *La moyenne de la satisfaction des deux acteurs et celle de la satisfaction de l'acteurMax en fonction de l'égocentrisme des deux acteurs.*

La figure 5.14 montre que plus l'égo-centrisme des acteurs diminue, plus l'algorithme converge vers une configuration élitiste avec une augmentation de la moyenne du maximum de la satisfaction des deux acteurs, et ce au détriment de la moyenne des satisfactions des deux acteurs. En effet, pour un égo-centrisme égale à 0,1, chacun des acteurs cherche à maximiser la satisfaction de l'acteur le plus satisfait quelque qu'il soit, et ce maximum n'est atteint que par la détérioration de la satisfaction de l'autre acteur. Les simulations convergent ainsi vers un état qui maximise la satisfaction de l'un et minimise celle de l'autre, ce qui produit une satisfaction moyenne des deux acteurs peu élevée.

Nous avons aussi considéré le modèle free-rider, où A1 utilise l'algorithme principal tandis que les trois autres acteurs sont élitistes. La figure 5.15 montre les résultats d'une analyse de sensibilité, comprenant 10 expériences où le paramètre égocentrisme est le même pour les trois acteurs (A2, A3 et A4) et varie entre 0,1 et 1. Les paramètres psycho-cognitifs sont initialisés de façon standard.

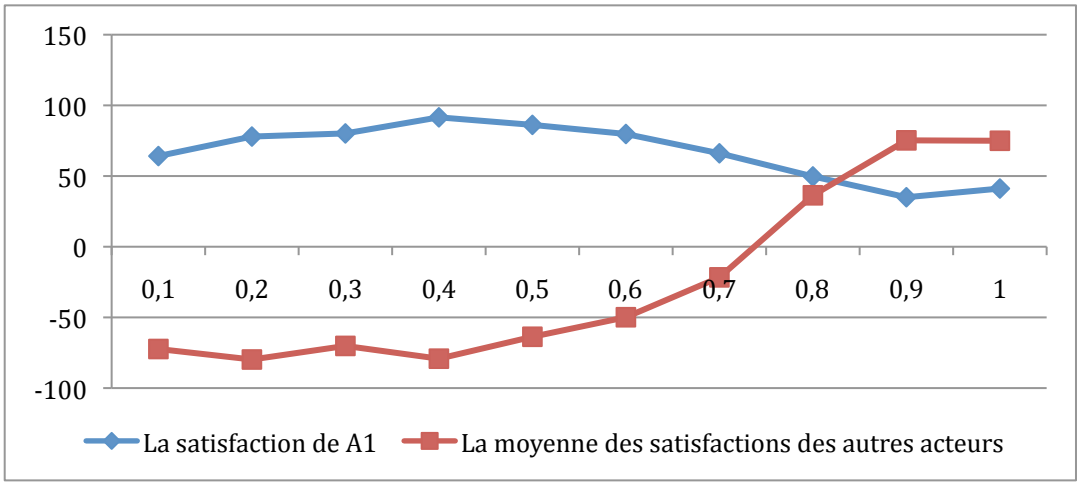


Figure 5.15. *La moyenne de la satisfaction de A1 et celle de la satisfaction moyenne des trois autres acteurs en fonction de l'égocentrisme de tous les acteurs.*

Plus l'égo-centrisme des trois acteurs A2, A3 et A4 diminue, plus ils s'intéressent et coopèrent afin d'augmenter la satisfaction de l'acteur le plus satisfait. La figure 5.15 montre que lorsque l'égo-centrisme diminue de 1 à 0,4, A1 profite de la coopération des autres acteurs et augmente largement sa satisfaction d'environ 40 à 90, au détriment de celle des autres. En deçà d'un égocentrisme de 0,4, les acteurs considèrent encore plus la satisfaction de l'acteurMax par rapport à leur propre satisfaction ; donc leur tendance à augmenter la satisfaction de l'acteurMax est donc plus forte, mais elle ne s'applique plus systématiquement à l'acteur A1. Plus l'égocentrisme diminue en deçà de 0,4, moins ils coopèrent avec A1 lorsqu'il n'est pas l'acteur le plus satisfait, et plus la satisfaction de A1 diminue.

5.3.5 – Analyse de sensibilité du paramètre égocentrisme / anti-élitiste

Le modèle d'organisation traité dans cette section a été construit pour montrer l'importance du paramètre égocentrisme dans l'algorithme anti-élitiste. Il comporte deux acteurs et deux relations (cf. tableau 5.18). Les enjeux des acteurs sur les relations et les fonctions d'effet sont définis comme montrés dans les tableaux 5.18 et 5.19.

	A1	A2
R1	**5**	4
R2	5	**6**

Tableau 5.18. *Les enjeux des acteurs sur les relations (les contours renforcés indiquent l'acteur contrôleur de la relation).*

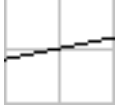
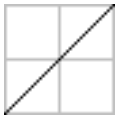
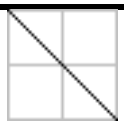
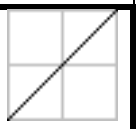
	A1	A2
R1		
R2		

Tableau 5.19. *Les fonctions d'effet des relations sur les acteurs.*

Tout d'abord, le tableau 5.20 présente les résultats de 200 simulations de l'algorithme principal sur ce modèle. Les paramètres psycho-cognitifs sont initialisés de façon standard : discernement = 1, ténacité = réactivité = 5, et une répartition de 50% (ou 60% en fonction de la somme des enjeux posés sur la relation que l'acteur contrôle) pour la dernière règle et 50% (ou 40%) pour l'avant dernière règle. 100% des simulations ont convergé en moins de 22 000 pas avec une moyenne de 20 500 pas.

	A1	A2
Satisfaction	-40	100
Ecart-type	0	0

Tableau 5.20. *Satisfaction des acteurs à l'issue de 200 simulations de l'algorithme principal sur un modèle à deux acteurs et deux relations.*

Les simulations convergent vers 10 pour l'état des deux relations R1 et R2, la configuration très élitiste correspondant au maximum global. En effet, la structure très simple du modèle fait en sorte que l'acteur A1 n'a pas le choix que de coopérer avec l'acteur A2, tandis que ce dernier profite de la coopération inconditionnelle de A1 et préserve l'impact de la relation qu'il contrôle. Cela résulte en une satisfaction maximale de A2, et une satisfaction proche du pire (-60) pour A1.

La figure 5.16 montre les résultats d'une analyse de sensibilité, comprenant 10 expériences où le paramètre égocentrisme de A1 varie entre 0,1 et 1 : A1 est anti-élitiste, tandis que A2 utilise l'algorithme principal. Les paramètres psycho-cognitifs sont initialisés de façon standard : discernement = 1, ténacité = réactivité = 5, et une répartition de 50% (ou 60% en fonction de la somme des enjeux posés sur la relation que l'acteur contrôle) pour la dernière règle et 50% (ou 40%) pour l'avant dernière règle.

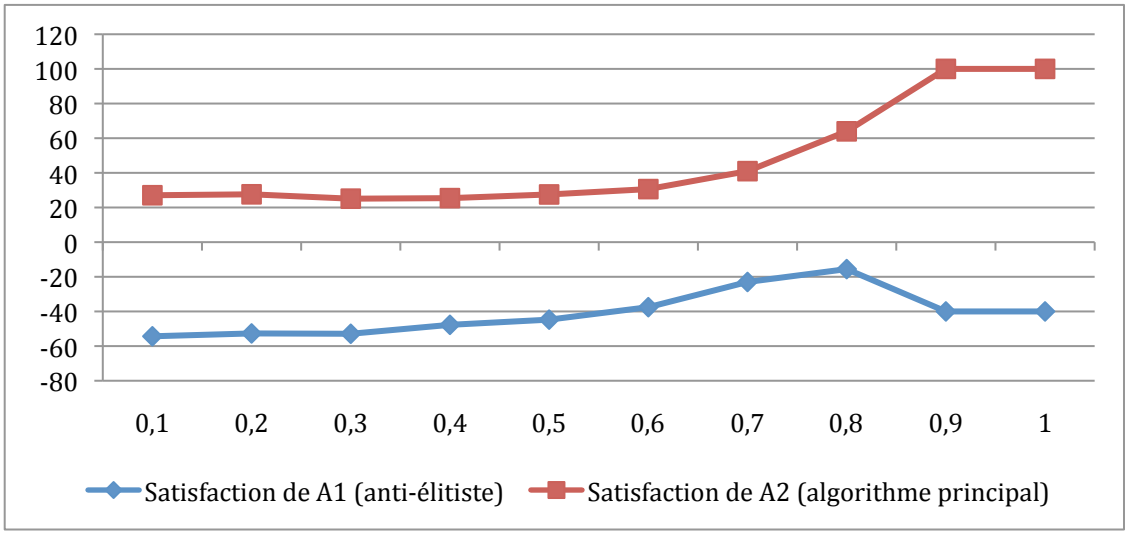


Figure 5.16. *La moyenne de la satisfaction de chaque acteur en fonction de l'égocentrisme de A1.*

La figure 5.16 montre que plus l'égocentrisme de A1 diminue, plus la satisfaction de A2 diminue. Tandis que la satisfaction de A1 augmente de -40 à environ -18 quand son égocentrisme diminue de 1 à 0,8, et elle diminue d'environ -15 à -55 quand il diminue de 0,8 à 0,1. En effet, pour un égocentrisme de 1 ou 0,9, A1 cherche à augmenter sa propre satisfaction et à légèrement faire diminuer celle de l'acteur le plus satisfait, A2. Les satisfactions de chaque acteur sont égales à celles obtenues par l'algorithme principal (cf. tableau 5.20). En deçà de 0,8, plus l'égocentrisme de A1 diminue, plus sa tendance à diminuer la satisfaction de A2 augmente, et plus il sacrifie l'impact sur lui-même de la relation qu'il contrôle afin de diminuer la satisfaction de A2. La figure 5.17

montre que plus l'égocentrisme de A1 diminuer, plus l'état de R1 diminue, et par conséquent plus l'impact de cette relation sur les deux acteurs diminue (cf. tableau 5.19).

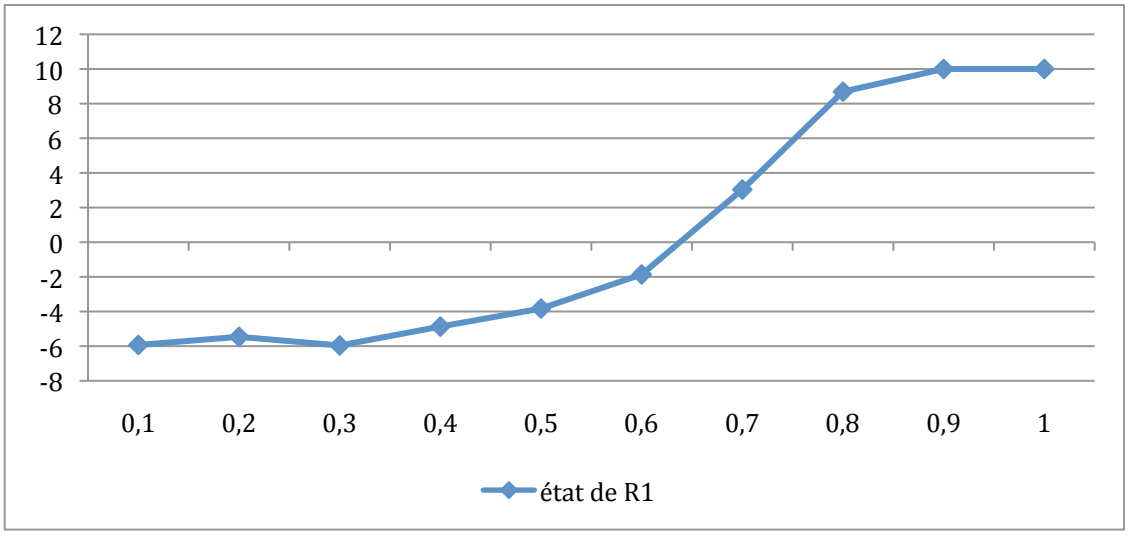


Figure 5.17. *La moyenne de l'état de la relation R1 en fonction de l'égocentrisme de A1.*

5.3.6 – Discussion

On ne prétend pas que les analyses présentées ci-dessus sont suffisantes pour évaluer les algorithmes élitiste et anti-élitiste, mais appliqué à deux ou trois modèles d'organisation, chaque algorithme donne des résultats conformes à ceux attendus.

L'algorithme élitiste conduit un acteur à participer plus ou moins en fonction de son égocentrisme à la réalisation de l'objectif de l'acteur le plus satisfait. D'une part, il ne faut pas s'attendre à ce que l'acteur élitiste arrive toujours, quelque soit son égocentrisme, à augmenter la satisfaction de l'acteur le plus satisfait. En effet, sa participation est contrainte par la structure de l'organisation, plus précisément par le pouvoir (cf. 2.1.4) qu'il exerce sur les autres, notamment sur l'acteurMax (cf. figure 5.15). D'autre part, un acteur élitiste n'augmentera pas nécessairement sa propre satisfaction ou la satisfaction globale de l'organisation, l'augmentation de la satisfaction de l'acteur en position d'élite se fait parfois au détriment d'un ou plusieurs acteurs. C'est ce qui se produit dans les cas du dilemme du prisonnier et du modèle free-rider présentés ci-dessus, où la maximisation de la satisfaction de l'acteurMax n'est possible que par la détérioration de la situation des autres. L'égocentrisme de l'acteur élitiste définit donc jusqu'à quel point il est prêt à accepter la détérioration de sa propre situation en faveur de celle de l'acteurMax.

Du même, un acteur anti-élitiste participe, s'il n'est pas le plus satisfait, à la diminution de la satisfaction de l'acteur le plus satisfait. Cette participation est fortement corrélée au pouvoir qu'il exerce sur l'acteurMax : plus son pouvoir augmente, plus il est en mesure de diminuer la satisfaction de cet acteur. dans certains cas, cette diminution peut être au détriment de la situation de l'acteur anti-élitiste. Son égocentrisme permet donc de définir jusqu'à quel point il est prêt à sacrifier sa propre situation afin de détériorer la situation de l'autre (cf. figures 5.16 et 6.17).

5.4 – Protectionnisme / anti-protectionnisme

Nous qualifions ici de protectionnisme (respectivement anti-protectionnisme) le comportement qui consiste à favoriser (respectivement défavoriser) les personnes jugées comme étant dans les situations les plus difficiles. Nous considérons donc ici que l'acteur à protéger est celui ayant la satisfaction minimale (on pourrait tout aussi bien considérer qu'il s'agit de l'acteur ayant la capacité d'action, le pouvoir ou l'influence minimal).

La rationalité de l'acteur protectionniste est tout à fait similaire à celle de l'acteur élitiste et ne diffère que par la cible de son attention : l'acteur le moins satisfait et non plus celui le plus satisfait. Un acteur protecteur perçoit donc à chaque pas de la simulation la satisfaction de l'acteur le moins satisfait et participe, plus ou moins en fonction de son égocentrisme, à la maximisation de sa satisfaction. De même, un acteur anti-protecteur cherche à empêcher l'acteur le moins satisfait de réaliser son objectif, c'est-à-dire tente de diminuer encore sa satisfaction.

5.4.1 – Présentation de l'algorithme

Cet algorithme, que l'on nomme *algorithme (anti-)protectionniste*, repose sur l'utilisation par les acteurs protectionnistes ou anti-protectionnistes d'informations sur l'acteur le moins satisfait.

- Un protectionniste calcule la variation de sa satisfaction en fonction de la variation de sa satisfaction personnelle, ΔsatisPerso (cf. éq. 4.17), et de celle de la satisfaction de l'acteurMin :

$$\Delta satisfaction_t\,(acteurP) = EC \times \Delta satisPerso_t\,(acteurP) + (1 - EC) \times \Delta satisfaction_t\,(acteurMin) \qquad \text{(éq. 5.19)}$$

  Un anti-protectionniste qui n'est pas l'acteur le moins satisfait cherche à diminuer la satisfaction de l'acteurMin ; il calcule donc la variation de sa satisfaction en fonction de la variation de sa propre satisfaction de laquelle il soustrait la variation de la satisfaction de l'acteurMin :

$$\Delta satisfaction_t\,(acteurAP) = EC \times \Delta satisPerso_t\,(acteurAP) - (1 - EC) \times \Delta satisfaction_t\,(acteurMin) \qquad \text{(éq. 5.20)}$$

  S'il est l'acteur le moins satisfait, il ne cherche pas à diminuer sa propre satisfaction, et donc[23] :

$$\Delta satisfaction_t\,(acteurAP) = \Delta satisPerso_t\,(acteurAP) \qquad \text{(éq. 5.21)}$$

- Un protectionniste calcule son écart à chaque pas de la simulation en fonction de son écart personnel (cf. éq. 4.4) et de l'écart de l'acteurMin :

$$écart_t\,(acteurP) = EC \times écartPerso_t\,(acteurP) + (1 - EC) \times écart_t\,(acteurMin) \qquad \text{(éq. 5.22)}$$

  Un anti-protectionniste qui n'est pas l'acteur le moins satisfait calcule son écart en fonction de son écart personnel et de l'opposé de celui de l'acteurMin :

$$écart_t\,(acteurAP) = EC \times écartPerso_t\,(acteurAP) + (1 - EC) \times (1 - écart_t\,(acteurMin)) \qquad \text{(éq. 5.23)}$$

  S'il est l'acteur le moins satisfait, il applique la formule suivante :

$$écart_t\,(acteurAP) = écartPerso_t\,(acteurAP) \qquad \text{(éq. 5.24)}$$

- Les autres variables de cet algorithme (l'ambition, le taux d'exploration, l'intensité des actions, et les forces des règles) sont calculées et mises à jour de la même façon que dans l'algorithme principal.

Un protectionniste ayant la satisfaction minimale est satisfait si sa propre satisfaction est supérieure où égale à son ambition ; dans le cas contraire, il n'est satisfait que si l'acteurMin est satisfait ($écart_t(acteurMin) \leq 0$). Par contre, un anti-protectionniste est satisfait si sa propre satisfaction est plus grande que son ambition. Comme dans le cas de l'élitisme, la dissymétrie entre les critères de satisfaction d'un acteur protectionniste et d'un acteur anti-protectionniste provient de ce que, pour ce dernier, il n'est pas possible de concilier en une seule grandeur la recherche d'une

[23] Bien évidemment, rien se s'oppose à qu'un acteur anti-protectionniste n'applique sa rationalité même lorsqu'il est le moins satisfait, l'équation 4.21 devenant alors $\Delta satisfaction_t\,(acteurAP) = -\,\Delta satisPerso_t\,(acteurAP)$

maximisation (sa satisfaction Personnelle) et celle d'une minimisation (la satisfaction de l'acteurMin).

Cet algorithme est similaire à l'algorithme (anti-)élitiste et ne diffère que par la cible de son attention : l'acteur le moins satisfait au lieu de celui le plus satisfait. Les étapes sont les suivantes :

Initiation :

L'état de chaque relation est initialisé arbitrairement à 0 (l'état neutre).

La satisfaction est calculée en fonction des états des relations dont l'acteur dépend (éq. 4.3)

L'ambition est initialisée à la valeur maximale de la satisfaction de l'acteur (éq. 4.6)

L'écart est calculé en fonction de la satisfaction et l'ambition (éq. 4.4 et 4.5)

Le taux d'exploration est initialisé à la valeur du taux d'exploration instantané (éq. 4.9, 4.10, et 4.11)

A chaque étape t de la simulation, chaque acteur :

1. perçoit sa satisfaction (éq. 4.3), calcule son écart (éq. 4.4, 5.22, 5.23 et 5.24), et met à jour son ambition (éq. 4.7 et 4.8).

2. met à jour son taux d'exploration (éq. 4.9, 4.10, 4.11, et 4.12).

3. met à jour l'intensité des actions (équation 4.13).

4. met à jour la force des deux dernières règles appliquées (éq. 4.15, 4.16, 4.17, 5.19, 5.20 et 5.21). Les règles de force négative sont oubliées.

5. sélectionne les règles applicables, celles dont la composante situation est proche de sa situation courante en fonction de son discernement (4.2).

6. Si l'ensemble des règles applicables est vide (notamment au début de la simulation), il crée une nouvelle règle, avec une force initialisée à 0, une situation égale à la situation courante et les actions (modifications à apporter sur les états des relations contrôlées) choisies au hasard dans l'intervalle [- intensité ; + intensité] (éq. 4.14).

7. choisit la règle nouvellement créée ou, parmi celles applicables, l'une de trois règles dont la force est la plus grande.

Lorsque tous les acteurs ont choisi une règle, leurs actions sont appliquées.

Algorithme 5.4. *Schéma de l'algorithme de rationalité des acteurs protecteurs et anti-protecteurs.*

5.4.2 – Exemples d'application / protectionniste

Afin d'évaluer la qualité de cet algorithme, nous analysons son comportement sur le cas Bolet, le modèle free-rider, et le cas Seita.

Le cas Bolet

Avec l'algorithme principal, le chef d'atelier est l'acteur défavorisé avec une satisfaction de -12,03 (cf. tableau 4.7), du fait d'un conflit structurel avec Jean-BE qui est résolu en faveur de ce dernier. Ces deux acteurs dépendent essentiellement de la relation contrôlée par le père. Pour étudier l'effectivité du protectionnisme de notre algorithme, nous considérons que le père est protectionniste avec un égocentrisme de 0,5, comportement en parfaite adéquation avec son rôle dans l'entreprise, tandis que les autres acteurs utilisent l'algorithme principal. Le tableau 5.21 présente les résultats de 200 simulations du cas Bolet ainsi que les résultats obtenus par l'algorithme principal. Les paramètres psycho-cognitifs sont initialisés de façon standard comme précédemment

en 5.2.1. Toutes les simulations ont convergé en moins de 10 000 pas avec une moyenne de 5 274 pas.

		CA	Père	André	Jean-BE	acteurMin	Satisfaction Globale
Algorithme protectionniste pour le père	Satisfaction	24,64	57,61	62,97	20,92	**20,22**	166,14
	Satis. en %	62,32%	93,36%	88,68%	62,9%	60,11%	90,57%
	Ecart-Type	5,34	2,54	6,24	6,3	6,4	
Algorithme Principal	Satisfaction	-12,03	56,98	43,61	46,97	**-12,05**	135,53
	Satis. en %	43,99%	92,85%	76,11%	78,96%	43,98%	82,77%
	Ecart-Type	3,19	1,08	3,38	3,49	3,17	

Tableau 5.21. *Satisfaction des acteurs à l'issue de 200 simulations du cas Bolet (père est protectionniste avec un égocentrisme de 0,5, tandis que les autres utilisent l'algorithme principal) ainsi que celles obtenues par l'algorithme principal.*

Remarquons tout d'abord que la satisfaction du père est quasiment égale à celle obtenue par l'algorithme principal, la satisfaction du CA et André est plus élevée , tandis que celle de Jean-BE est moindre que celle obtenue par l'algorithme principal. La satisfaction globale augmente.

Donnons une interprétation à ces résultats. Le tableau 5.22 montre la moyenne de l'état de chaque relation à l'issue des simulations présentées dans le tableau 5.21. Le père contrôle la décision de l'achat (DA) de laquelle les deux acteurs (CA et Jean-BE) dépendent fortement avec 4 points d'enjeux et des fonctions d'effet de sens opposés. Etant protectionniste et égo-centré à 0,5, il cherche à augmenter la satisfaction de l'acteur le moins satisfait, le chef d'atelier, autant qu'à augmenter la sienne. Il équilibre donc l'état de la relation qu'il contrôle à -0,35, de façon à augmenter le minimum des impacts de cette relation sur les quatre acteurs ; ce qui permet d'augmenter largement la satisfaction du CA, et diminue celle des autres acteurs. Le chef d'atelier, détectant la coopération du Père, coopère à son tour en plaçant la relation qu'il contrôle (application-prescription, AP) dans un état qui satisfait les autres acteurs. Les simulations convergent ainsi vers un état qui procure une « bonne » satisfaction à l'acteur le moins satisfait.

		Relation					
		DA	AP	IDP	CAP	NP	CNP
Algorithme protectionniste pour le Père	État final	**-0,35**	**4,69**	7,89	0,52	5,56	0,59
	Ecart-Type	0,61	4,67	1,64	2,01	3,11	1,61
Algorithme Principal	État final	**8,57**	**-8,44**	9,63	-0,38	8,77	1,35
	Ecart-Type	1,53	1,92	0,51	1,48	1,26	1,35

Tableau 5.22. *État des relations à l'issue de 200 simulations du cas Bolet (CA, Jean-BE et André utilisent l'algorithme principal, tandis que le père est protectionniste avec un égocentrisme de 0,5) ainsi que ceux obtenus par l'algorithme principal.*

Le modèle free-rider

Le cas traité dans cette analyse est le modèle free-rider où tous les acteurs sont protectionnistes et devraient donc assurer à A1, qui est le moins satisfait avec l'algorithme principal, une meilleure situation. Le tableau 5.23 présente les résultats de 200 simulations de ce

modèle ainsi que les résultats obtenus par l'algorithme principal. Les paramètres psycho-cognitifs et psycho-social sont initialisés de façon standard comme précédemment. 88% des simulations ont convergé en moins de 100 000 pas avec une moyenne de 1 842 pas.

		A1	A2	A3	A4	Satisfaction Globale
Algorithme protectionniste	Satisfaction	79,55	79,57	79,55	79,49	318,16
	Satisfaction en %	89,78 %	89,79 %	89,78 %	89,75 %	99,71 %
	Ecart-Type	0,79	0,72	0,74	0,85	
Algorithme Principal	Satisfaction	25,1	85,9	86,3	86,1	283,4
	Satisfaction en %	62,55 %	92,95 %	93,15 %	93,05 %	94,28 %
	Ecart-Type	11,53	8,32	8,63	8,48	

Tableau 5.23. *Satisfaction des acteurs à l'issue de 200 simulations du modèle free-rider où tous les acteurs sont protectionnistes ainsi que celles obtenues avec l'algorithme principal.*

Les résultats de simulations obtenus avec l'algorithme protectionniste favorisent la situation de A1. La configuration C1 (ou une configuration proche du C1), correspondant au maximum de la satisfaction globale, est atteinte dans 100% des simulations, alors qu'il ne l'est que dans 10,5% des simulations avec l'algorithme principal. En effet, les acteurs protectionnistes tentent d'améliorer la satisfaction de l'acteur le moins satisfait, A1 dans la plupart des cas. Ce dernier profite de cette coopération et améliore sa propre satisfaction sans chercher à empêcher les autres d'améliorer leur satisfaction. 88% des simulations ont convergé rapidement vers un état proche du maximum global (configuration C1), tandis que dans les autres simulations, les satisfactions des acteurs sont bloqués dans un intervalle proche de l'optimal et s'alternent comme illustré dans la figure 5.18 sans converger même après 100 000 pas (environ 54 fois la moyenne du nombre de pas).

Ces résultats sont les mêmes que ceux obtenus dans le cas où A1 utilise l'algorithme principal tandis que les trois autres acteurs sont anti-élitistes avec un égocentrisme de 0,5 (cf. le tableau 5.15 en 5.3.3 ci-dessus) : soutenir A1 (protection) revient à empêcher la défection de A1, A2 et A3 (anti-élitisme) et réciproquement.

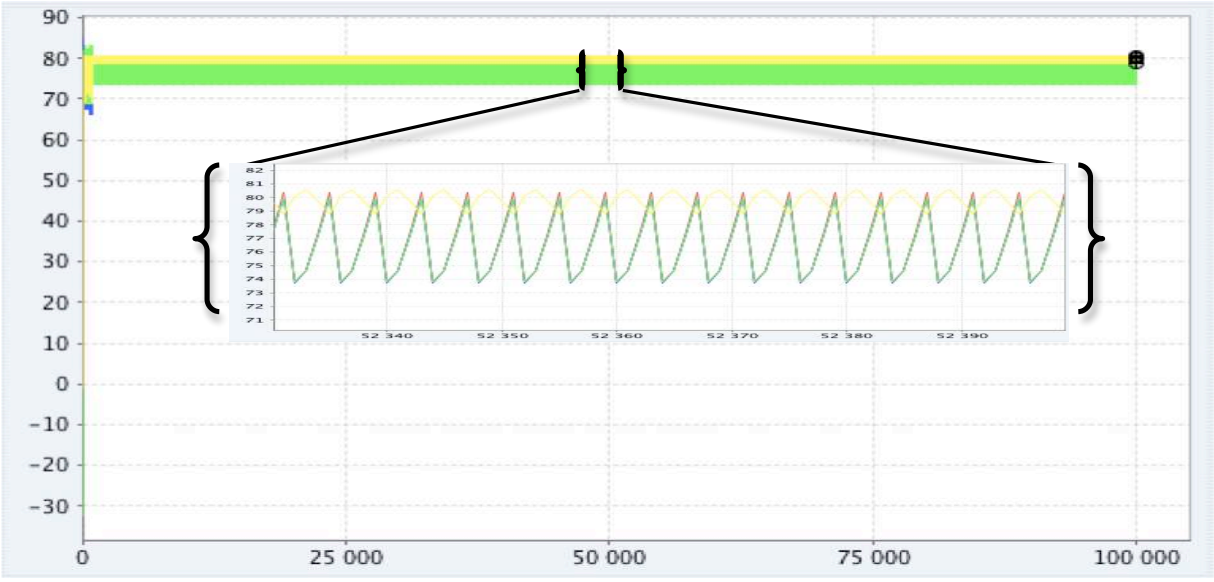


Figure 5.18. *La satisfaction des quatre acteurs au cours de la 16ème simulation.*

5.4.3 – Exemples d'application / anti-protectionniste

Nous analysons le comportement de l'algorithme anti-protectionniste sur les mêmes modèles d'organisation : le cas Bolet et le modèle free-rider.

Le cas Bolet

Dans cette section, nous considérons que le père et André sont anti-protectionnistes avec un égocentrisme de 0,5, tandis que CA et Jean-BE utilisent l'algorithme principal. On s'attend donc à ce que le père et/ou André sacrifient leur propre satisfaction afin de minimiser la satisfaction de l'un des deux acteurs structurellement défavorisés, le chef d'atelier ou Jean-BE.

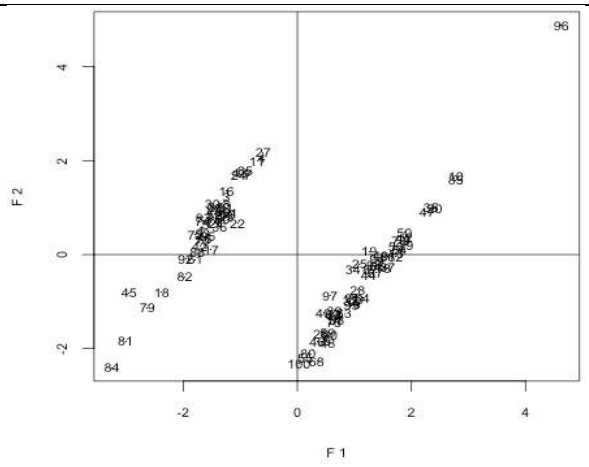
Le tableau 5.24 présente les résultats de 200 simulations du cas Bolet. Les paramètres psycho-cognitifs sont initialisés comme précédemment. Toutes les simulations ont convergé en moins de 11 000 pas avec une moyenne de 8 904 pas.

		CA	Père	André	Jean-BE	acteurMin
satisfaction	Moyenne	22,63	54,63	44,61	7,49	**-28,45**
	Maximum	80	62,93	66,44	67,53	-6,06
	Minimum	**-44,2**	42,03	1,09	**-42,56**	-44,2
	Ecart-Type	46,89	2,32	6,3	38,61	5,4

Tableau 5.24. *Satisfaction des acteurs et celle de l'acteur le moins satisfait (acteurMin) à l'issue de 200 simulations du cas Bolet (CA et Jean-BE utilisent l'algorithme principal, tandis que le père et André sont anti-protectionnistes avec un égocentrisme de 0,5).*

Remarquons tout d'abord que ce sont les deux acteurs CA et Jean-BE qui sont les moins satisfaits en moyenne : la satisfaction du père ne descend pas en dessous de 42,03 et e celle d'André en dessous de 1,09. Tandis que la satisfaction du CA varie entre -44,2 et 80 avec une moyenne de 22,63 et celle du Jean-BE entre -42,56 et 67,53 avec une moyenne de 7,49, tout deux avec un écart type important.

La figure 5.19 montre une analyse en composantes principales des résultats de simulations présentés dans le tableau 5.24.



(a) représentation graphique des résultats de simulations (en ne considérant que la satisfaction des acteurs) dans un repère formé des deux axes associés aux deux premières composantes principales (F1 et F2)

(b) contribution des variables (les satisfactions des quatre acteurs) aux deux premières composantes principales (F1 et F2). On pourra remarquer que André apparait en situation parfaite d'arbitrage entre Jean_BE et CA.

Figure 5.19. *Analyse en composantes principales des résultats de simulation du cas Bolet obtenue avec le logiciel R (98,21 de la variance est expliquée).*

Le père et André contrôlent 60% de la réalisation de l'objectif du CA et de Jean-BE, ils disposent donc du pouvoir leur permettant de diminuer de façon sélective la satisfaction de CA ou de Jean. En effet, l'état qui fait diminuer la satisfaction de l'un est bénéfique à l'autre, ce qui mène à une satisfaction moyenne peu élevée pour les deux acteurs CA et Jean-BE.

Une Classification Ascendante Hiérarchique permet de caractériser les deux modes vers lesquels les simulations convergent. Le tableau 5.25 montre la moyenne de l'état des relations et de la satisfaction des acteurs pour chacun des modes, et les figures 5.20 et 5.21 les boxplots des relations et des acteurs dans chacun des modes. Il apparaît que ce sont le Père (relation Decision-achat) et André (relations Contrôle-application-presc et Contrôle-nature-presc) qui changent de comportement d'un mode à un autre afin de minimiser la satisfaction de l'acteur le moins satisfait, le chef d'atelier (45% des simulations) ou Jean-BE (55% des simulations).

		Mode 1	Mode 2
Effectifs		45	55
Nombre de pas		18276	19491
État des relations	Decision-achat	7,98	-7,7
	Application-prescription	-6,9	-6,24
	Investissement-dans-prod	9,45	9,41
	Contrôle-application-presc	5,35	-0,62
	Nature-prescription	7,9	7,11
	Contrôle-nature-presc	0,24	6,03
Satisfactions	CA	-29,48	65,26
	Père	56,35	53,23
	André	43,25	45,72
	Jean-BE	50,4	-27,61

Tableau 5.25. *Moyenne de l'état des relations et la satisfaction des acteurs dans les deux modes.*

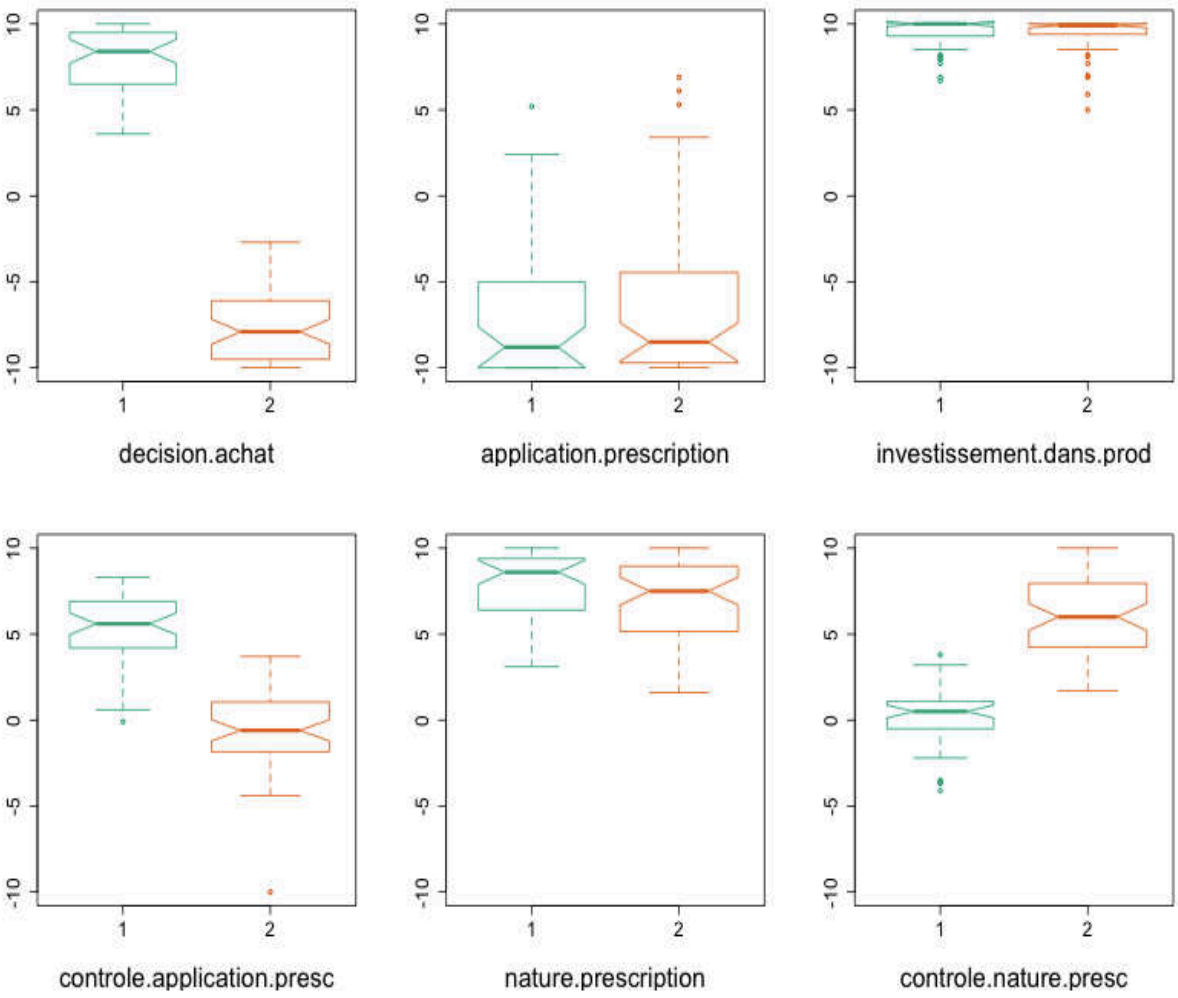


Figure 5.20. *Boxplot de l'état des relations dans chacun des deux modes.*

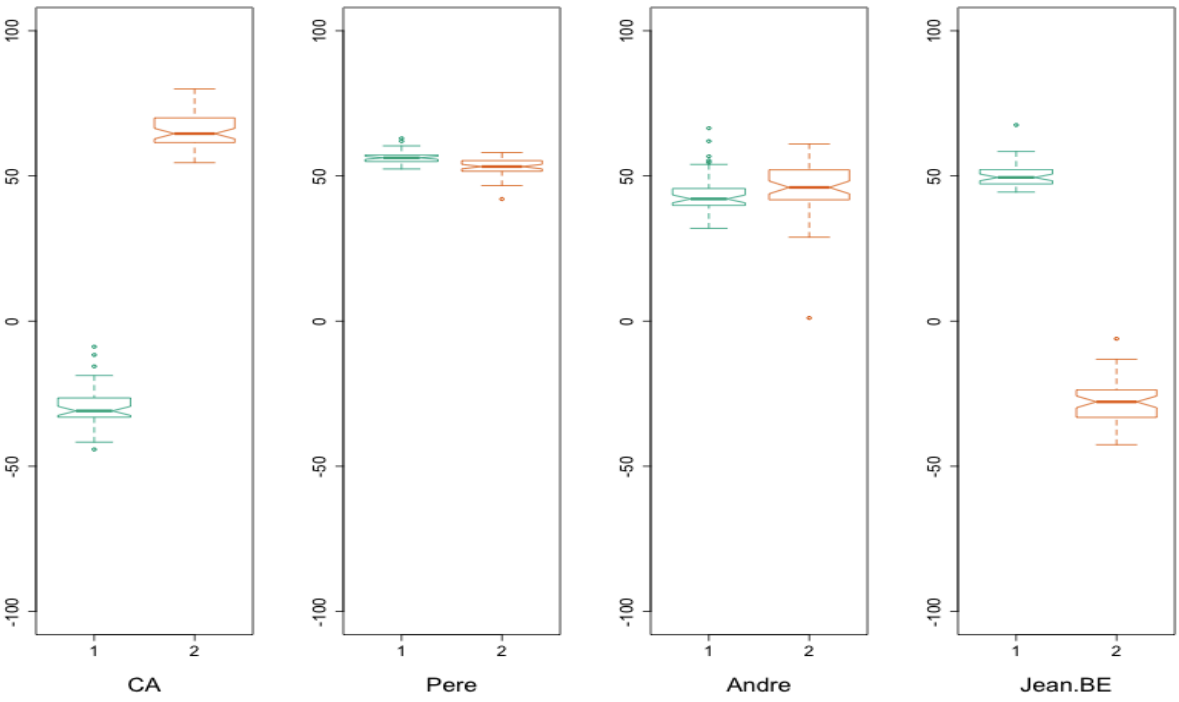


Figure 5.21. *Boxplot de la satisfaction des acteurs dans chacun des deux modes.*

Le modèle free-rider

Le cas traité dans cette analyse est le modèle free-rider où A1 utilise l'algorithme principal et les autres acteurs sont anti-protectionnistes. On s'attend à ce que les trois acteurs se liguent contre l'acteur A1 (l'acteur le plus moins) afin de minimiser sa satisfaction.

Les tableaux 5.26 et 5.27 présentent les résultats de 200 simulations de ce modèle ainsi que les résultats obtenus par l'algorithme principal. Les paramètres psycho-cognitifs sont initialisés de façon standard. Toutes les simulations ont convergé en moins de 17 000 avec une moyenne de 8 375 pas.

		A1	A2	A3	A4	acteurMin	Satisfaction Globale
Algorithme anti-protectionniste pour A2, A3 et A4	Satisfaction	7,4	88,1	88,6	87,5	**7,4**	271,6
	Satis. en %	53,7%	94,05%	94,3%	93,75%	53,7%	92,44
	Ecart-Type	19,91	9,64	9,8	9,38	19,91	
Algorithme Principal pour les quatre acteurs	Satisfaction	25,1	85,9	86,3	86,1	**25,1**	283,4
	Satis. en %	62,55%	92,95%	93,15%	93,05%	62,55%	94,28 %
	Ecart-Type	11,53	8,32	8,63	8,48	11,53	

Tableau 5.26. *Satisfaction des acteurs à l'issue de 200 simulations du modèle free-rider où A1 utilise l'algorithme principal tandis que les autres acteurs sont anti-protectionnistes ainsi que les résultats obtenus avec l'algorithme principal pour les quatre acteurs.*

	Configurations	C1	C2	C3	C4	C5	C6	C7	C8	C9
Algorithme anti-protectionniste A2, A3 et A4	% d'occurrences	0	26,5	28,5	24	7,5	6,5	7	0	0
Algorithme Principal pour les quatre acteurs		10,5	29,5	30,5	29,5	0	0	0	0	0

Tableau 5.27. *Le pourcentage d'apparition de chacun des neuf états dans les résultats de simulations lorsque A1 utilise l'algorithme principal tandis que les autres acteurs sont anti-protectionnistes, et ceux obtenus avec l'algorithme principal pour les quatre acteurs : l'état qui maximise la satisfaction globale (C1) ; les quatre états qui maximisent la satisfaction de l'un des quatre acteurs : A1 (C8), A2 (C2), A3 (C3) et A4 (C4) ; trois états dans lesquels deux des trois acteurs ne coopèrent pas : A2 et A3 (C5), A2 et A4 (C6), A3 et A4 (C7) ; l'équilibre de Nash (C9).*

Les résultats de simulations obtenus dans cette simulation pénalisent largement l'acteur A1 au bénéfice des acteurs A2, A3 et A4. La configuration C1, correspondante au maximum de la satisfaction globale, n'est atteinte dans aucune simulation, alors qu'elle l'était dans 10,5% des simulations avec l'algorithme principal. Tandis que les configurations C5, C6 et C7, dans lesquelles deux acteurs sur les trois (A2, A3 et A4) profitent de la coopération d'un seul parmi eux et préservent l'impact de la relation qu'ils contrôlent, sont atteintes dans 21% des simulations (les plus longues), alors qu'elles ne l'étaient dans aucune avec l'algorithme principal, comme s'il fallait que les simulations durent plus longtemps afin qu'ils perçoivent qu'ils puissent diminuer encore la satisfaction de A1. Dans les autres simulations (79% des simulations, les plus courtes), au moins un des trois acteurs (A2, A3 et A4) préserve l'impact de la relation qu'il contrôle, ce qui fait diminuer la satisfaction de A1.

L'analyse de sensibilité sur le modèle free-rider dans 5.4.5 montre que plus l'égocentrisme des acteurs diminue, plus ils auront tendance à diminuer la satisfaction de A1.

5.4.4 – Analyse de sensibilité du paramètre égocentrisme / protectionnisme

Le modèle d'organisation considéré dans cette section est le cas Seita où MainE, en position d'avantage structurel très marqué, est protectionniste, tandis que les deux autres acteurs utilisent l'algorithme principal. La figure 5.22 montre les résultats d'une analyse de sensibilité, comprenant 10 expériences où le paramètre égocentrisme du MainE varie entre 0,1 et 1. Les paramètres psycho-cognitifs sont initialisés de façon standard avec une répartition de 30% (ou 20% ou 50% en fonction de la somme des enjeux posés sur la relation que l'acteur contrôle) pour la dernière règle et 70% (ou 80% ou 50%) pour l'avant dernière règle.

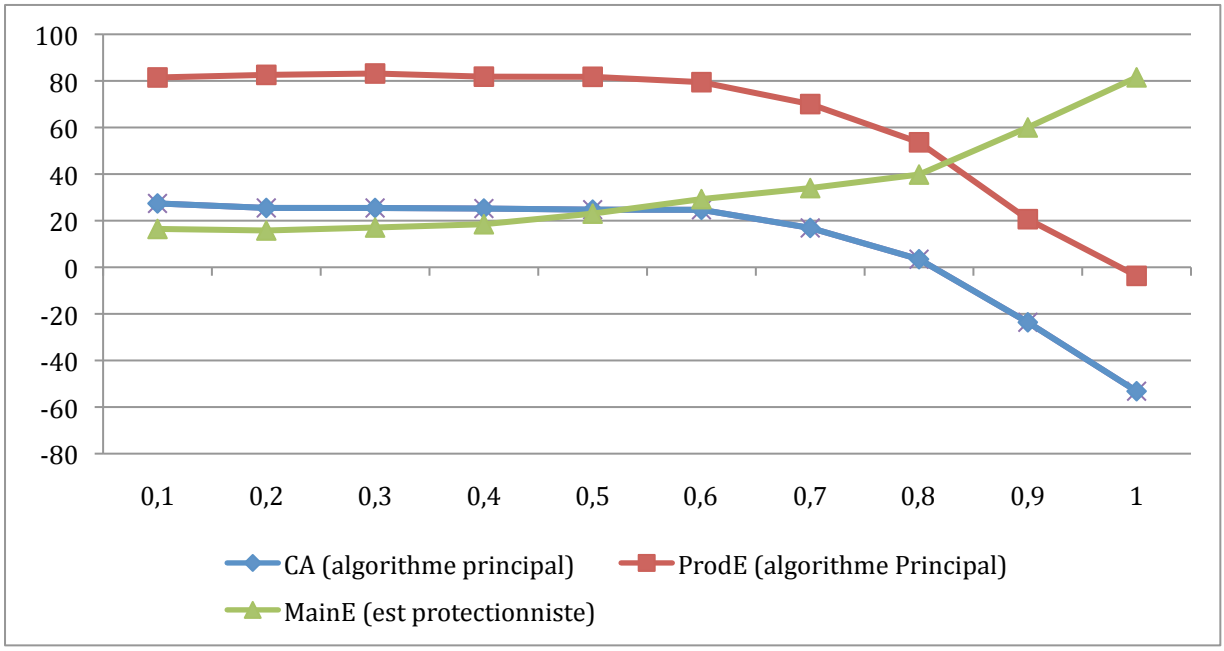


Figure 5.22. *La moyenne de la satisfaction des trois acteurs en fonction de l'égocentrisme de MainE.*

La figure 5.22 montre que plus les ouvriers de maintenance (MainE) sont protectionnistes (avec la diminution de leur égocentrisme), plus ils sont effectivement protecteur et s'améliorent la satisfaction des acteurs les moins satisfaits, CA et ProdE. En effet, les ouvriers de maintenance disposent d'un pouvoir leur permettant de diriger le jeu comme ils veulent : ils contrôlent 50% de ce dont ils dépendent, et les relations qu'ils contrôlent imposent des contraintes aux relations contrôlées par les autres acteurs. Cet effet du protectionnisme est très sensible pour une variation de l'égocentrisme de 1 à 0,5, mais pas en deçà de cette valeur.

5.4.5 – Analyse de sensibilité du paramètre égocentrisme / anti-protectionnisme

Nous reprenons dans cette section le modèle free-rider où A1 utilise l'algorithme principal, tandis que les trois autres acteurs sont anti-protectionnistes et donc se liguent contre lui. Les figures 5.23 et 5.24 montrent les résultats d'une analyse de sensibilité où le paramètre égocentrisme est le même pour les acteurs A2, A3, et A3 et varie entre 0,1 et 1. Les paramètres psycho-cognitifs étant initialisés de façon standard.

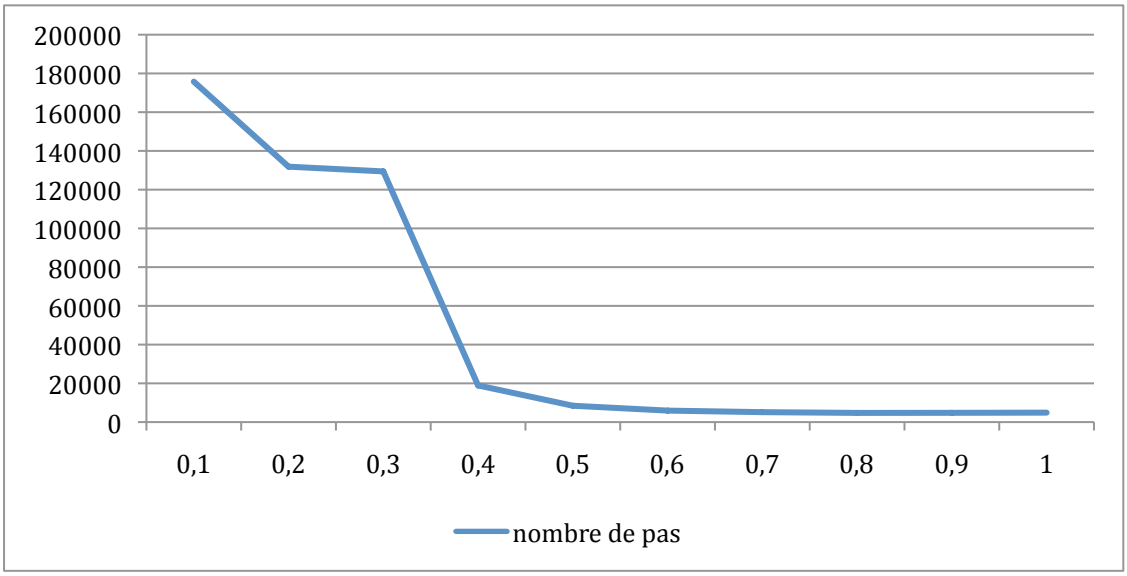


Figure 5.23. *Le nombre de pas moyen pour les simulations qui ont convergé, en fonction de l'égocentrisme des acteurs A2, A3, et A4 dans le modèle free-rider.*

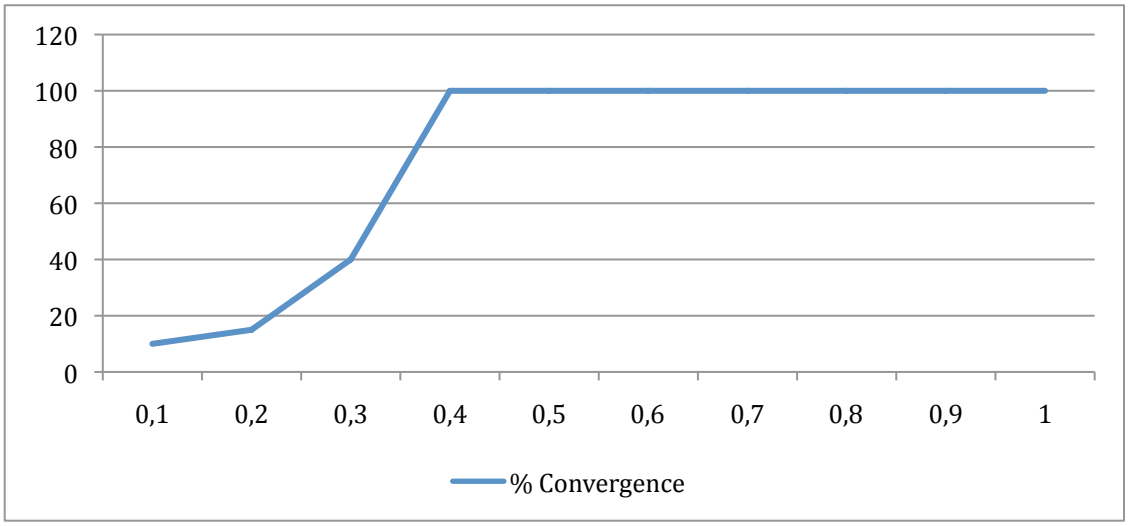


Figure 5.24. *Le pourcentage de convergence des simulations en moins de 300 000 pas en fonction de l'égocentrisme des acteurs A2, A3, et A4 dans le modèle free-rider.*

La figure 5.25 montre que plus l'égocentrisme des trois acteurs diminue, plus ils s'acharnent à la détérioration de la situation de l'acteur le moins satisfait (A1), et plus les simulations durent longtemps (cf. figure 5.23). En deçà de 0,4, le pourcentage de convergence des simulations en moins de 300 000 pas diminue de 100% à 10% pour un égocentrisme de 0,1 (cf. figure 5.24).

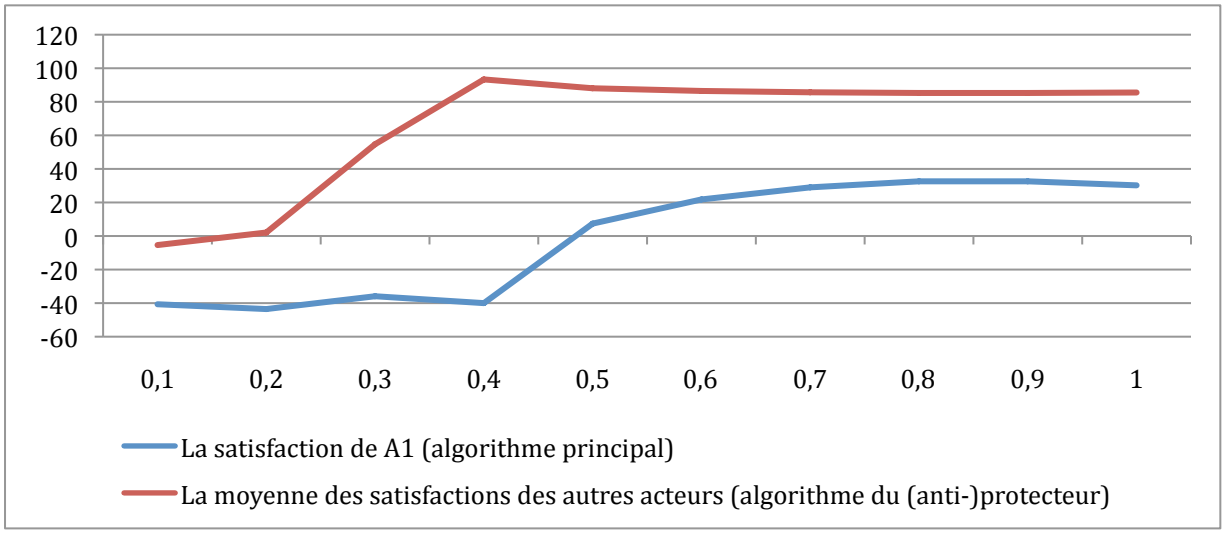


Figure 5.25. *La moyenne de la satisfaction de A1 et celle de la satisfaction moyenne des trois autres acteurs en fonction de l'égocentrisme des acteurs A2, A3, et A4.*

En ce qui concerne la figure 5.25, plus l'égocentrisme des trois acteurs diminue, plus leur tendance à détériorer la situation de l'acteur le moins satisfait (A1) augmente même si cela les pénalise largement du fait de la non coopération de A1.

L'effet de la variation de l'égocentrisme des acteurs se fait donc sentir sur la plage de valeur de 0,1 à 0,6 et il est nul au-delà.

5.4.6 – Discussion

En ce qui concerne l'algorithme protectionniste, dans les deux cas présentés ci-dessus, si l'acteur qui contrôle la satisfaction de l'acteur le moins satisfait (A1 dans le modèle free-rider, et CA et Jean-BE dans le cas Bolet) adopte un comportement protectionniste, il va effectivement sensiblement améliorer la situation de l'acteur le moins satisfait. Cela peut se produire au détriment de l'acteur protectionniste, comme dans le cas Seita où l'augmentation de la satisfaction du chef d'atelier et des ouvriers de production pénalise largement les ouvriers de maintenance (cf. figure 5.22). L'égocentrisme d'un acteur protectionniste détermine donc jusqu'à quel point un acteur acceptera la détérioration de sa situation en faveur de celle de l'acteur le moins satisfait.

De même un anti-protectionniste ayant un pouvoir structurel sur l'acteur le moins satisfait va largement faire diminuer sa satisfaction. Par exemple, dans le cas Bolet où le père et André coordonnent leurs actions afin de diminuer la satisfaction de l'acteur le moins satisfait, à savoir Jean-BE dans 55% des cas et le chef d'atelier dans les autres cas. Dans le modèle free-rider, A1 est toujours le moins satisfait car l'amélioration de sa satisfaction nécessité la coopération des trois autres acteurs en même temps. Étant anti-protectionnistes, ces trois acteurs coordonnent leurs actions afin de diminuer encore la satisfaction de A1. Cette coordination d'actions contre A1 leur coûte très cher dès que leur égocentrisme diminue en deçà de 0,4 du fait qu'alors, A1 garde pour lui-même l'impact de la relation qu'il contrôle et diminue la satisfaction des autres (cf. figure 5.25).

5.5 – Égalitariste / anti Égalitariste

L'égalitarisme est le comportement qui cherche à réduire l'écart entre les situations des acteurs. Nous considérons ici que le rétrécissement de l'éventail des situations des acteurs peut être obtenu par l'attitude conjointe de l'anti-élitisme et du protectionnisme. En d'autres termes, il s'agit de réduire l'écart entres les élites et les défavorisés, c'est-à-dire entre l'acteur le plus satisfait et celui le moins satisfait. Quant à l'anti-égalitarisme, c'est le comportement inverse qui cherche à agrandir cet écart.

Dans les rationalités présentées ci-dessous, un acteur égalitariste perçoit à chaque pas de la simulation les satisfactions de l'acteur le moins satisfait et de celui le plus satisfait, et il cherche à réduire l'écart entre les deux, en participant à la diminution de la satisfaction de l'acteur le plus satisfait et à l'augmentation de la satisfaction de l'acteur le moins satisfait. Un anti-égalitariste cherchera inversement à agrandir l'écart entre les deux acteurs le moins et le plus satisfait.

5.5.1 – Présentation de l'algorithme

Cet algorithme, que l'on nomme *algorithme (anti-)égalitariste*, repose sur l'utilisation d'informations sur les deux acteurs le moins satisfait, acteurMin, et le plus satisfait, acteurMax. Un acteur égalitariste « acteurEg » ou anti-égalitariste « acteurAEg » doit être en mesure d'évaluer la situation de ces deux acteurs, afin de participer à la réduction (respectivement l'agrandissement) de l'écart entre eux, en agissant de la façon suivante :

- Un égalitariste qui n'est pas l'acteur le plus satisfait cherche à augmenter la satisfaction de l'acteurMin et à diminuer celle de l'acteurMax ; il calcule la variation de sa satisfaction en fonction de la variation de sa satisfaction personnelle, ΔsatisPerso (cf. éq. 4.17), à laquelle il ajoute la variation de la satisfaction de l'acteurMin et soustrait la variation de la satisfaction de l'acteurMax :

$$\Delta satisfaction_t\,(acteurEg) = EC \times \Delta satisfaction_t\,(acteurEg) + (1 - EC) \times (\Delta satisfaction_t\,(acteurMin) - \Delta satisfaction_t(acteurMax))\,/2 \quad \text{(éq. 5.25)}$$

S'il est l'acteur le plus satisfait, il ne cherche pas à diminuer sa propre satisfaction, et donc :

$$\Delta satisfaction_t\,(acteurEg) = EC \times \Delta satisfaction_t\,(acteurEg) + (1 - EC) \times \Delta satisfaction_t\,(acteurMin) \quad \text{(éq. 5.26)}$$

Un anti-égalitariste qui n'est pas l'acteur le moins satisfait cherche à augmenter la satisfaction de l'acteurMax et à diminuer celle de l'acteurMin ; il calcule la variation de sa satisfaction en fonction de la variation de sa satisfaction personnelle à laquelle il ajoute la variation de la satisfaction de l'acteurMax et soustrait la variation de la satisfaction de l'acteurMin :

$$\Delta satisfaction_t\,(acteurEg) = EC \times \Delta satisfaction_t\,(acteurEg) + (1 - EC) \times (\Delta satisfaction_t\,(acteurMax) - \Delta satisfaction_t(acteurMin))\,/2 \quad \text{(éq. 5.27)}$$

S'il est l'acteur le moins satisfait, il ne cherche pas à diminuer sa propre satisfaction, et donc :

$$\Delta satisfaction_t\,(acteurEg) = EC \times \Delta satisfaction_t\,(acteurEg) + (1 - EC) \times \Delta satisfaction_t\,(acteurMax) \quad \text{(éq. 5.28)}$$

- Un égalitariste qui n'est pas l'acteur le plus satisfait calcule son écart à chaque instant en fonction de son écart personnel (cf. éq. 4.4), de l'écart de l'acteurMin et de l'opposé de l'écart de l'acteurMax :

$$écart_t\,(acteurEg) = EC \times écart_t\,(acteurEg) + (1 - EC) \times (écart_t\,(acteurMin) + (1 - écart_t(acteurMax)))\,/2 \quad \text{(éq. 5.29)}$$

S'il est l'acteur le plus satisfait, il applique la formule suivante :

$$écart_t\,(acteurEg) = EC \times écart_t\,(acteurEg) + (1 - EC) \times écart_t\,(acteurMin) \quad \text{(éq. 5.30)}$$

Un anti-égalitariste qui n'est pas l'acteur le moins satisfait, contrairement à l'égalitariste, calcule son écart à chaque instant en fonction de son écart personnel, de l'écart de l'acteurMax et de l'opposé de l'écart de l'acteurMin :

$$\text{écart}_t(\text{acteurEg}) = EC \times \text{écart}_t(\text{acteurEg}) + (1 - EC) \times (\text{écart}_t(\text{acteurMax}) + (1 - \text{écart}_t(\text{acteurMin}))) / 2 \quad \text{(éq. 5.31)}$$

S'il est l'acteur le moins satisfait, il applique la formule suivante :

$$\text{écart}_t(\text{acteurEg}) = EC \times \text{écart}_t(\text{acteurEg}) + (1 - EC) \times \text{écart}_t(\text{acteurMax}) \quad \text{(éq. 5.32)}$$

- Les autres variables de cet algorithme (l'ambition, le taux d'exploration, l'intensité des actions, et les forces des règles) sont calculées et mises à jour de la même façon que dans l'algorithme principal.

Un acteur égalitariste n'est satisfait que si l'acteurMin est satisfait ($\text{écart}_t(\text{acteurMin}) <= 0$). En effet, il est impossible de concilier en une seule grandeur la recherche d'une maximisation (la satisfaction de l'acteurMin) et celle d'une minimisation (la satisfaction de l'acteurMax). De même, un acteur anti-égalitariste n'est satisfait que si l'acteurMax est satisfait ($\text{écart}_t(\text{acteurMax}) <= 0$).

Les étapes de l'algorithme sont les suivantes :

Initiation :

L'état de chaque relation est initialisé arbitrairement à 0 (l'état neutre).

La satisfaction est calculée en fonction des états des relations dont l'acteur dépend (éq. 4.3)

L'ambition est initialisée à la valeur maximale de la satisfaction de l'acteur (éq. 4.6)

L'écart est calculé en fonction de la satisfaction et l'ambition (éq. 4.4 et 4.5)

Le taux d'exploration est initialisé à la valeur du taux d'exploration instantané (éq. 4.9, 4.10, et 4.11)

A chaque étape t de la simulation, l'acteur :

1. perçoit sa satisfaction (éq. 4.3), calcule son écart (éq. 4.4, 5.25, 5.26, 5.27 et 5.28), et met à jour son ambition (éq. 4.7 et 4.8).

2. met à jour son taux d'exploration (éq. 4.9, 4.10, 4.11, et 4.12).

3. met à jour l'intensité des actions (équation 4.13).

4. met à jour la force des deux dernières règles appliquées (éq. 4.15, 4.16, 4.17, 5.21, 5.22, 5.23 et 5.24). Les règles de force négative sont oubliées.

5. sélectionne les règles applicables, celles dont la composante situation est proche de sa situation courante en fonction de son discernement (4.2).

6. Si l'ensemble des règles applicables est vide (notamment au début de la simulation), il crée une nouvelle règle, avec une force initialisée à 0, une situation égale à la situation courante et les actions (modifications à apporter sur les états des relations contrôlées) choisies au hasard dans l'intervalle [- intensité ; + intensité] (éq. 4.14).

7. choisit la règle nouvellement créée ou, parmi celles applicables, l'une de trois règles dont la force est la plus grande.

Lorsque tous les acteurs ont choisi une règle, leurs actions sont appliquées.

Algorithme 5.5. *Schéma de l'algorithme de rationalité des acteurs égalitariste et anti-égalitaristes.*

5.5.2 – Exemples d'application

Dans cette section, nous présentons les résultats de cet algorithme sur un modèle à trois acteurs et quatre relations.

Un modèle à trois acteurs et quatre relations / égalitariste

Le modèle d'organisation traité dans cette section comporte trois acteurs et quatre relations (cf. tableau 5.28). Les enjeux des acteurs sur les relations et les fonctions d'effet sont définis comme illustré dans les tableaux 5.28 et 5.29. Ce modèle a été construit pour mettre en évidence la différence entre l'égalitarisme d'une part, et l'anti-élitisme et le protectionnisme d'autre part. En effet, ce modèle présente un conflit structurel entre A1 et A2 sur la relation R21 contrôlée par A2, et une similarité d'intérêt entre A2 et A3,leurs fonctions d'effet sur les relations R22 et R3 variant dans le même sens. A2 est l'acteur le plus influent, avec 14 points d'enjeux placés sur les relations qu'il contrôle. Si A2 est protectionniste, on s'attend à ce qu'il sacrifie l'impact de la relation qu'il contrôle (R21) pour participer à l'augmentation de la satisfaction de A1. Par contre s'il est anti-élitiste, on s'attend à ce qu'il lutte contre l'acteur A3 et équilibre l'état de la relation R22 de façon d'être lui-même l'acteur le plus satisfait. Finalement s'il est égalitariste, il doit réduire l'écart entre la satisfaction des acteurs, et donc augmenter la satisfaction de A1 et diminuer celle de A3.

	A1	A2	A3
R1	**5**	3	0
R21	5	**3**	0
R22	0	**1**	5
R3	0	3	**5**

Tableau 5.28. *Les enjeux des acteurs sur les relations (les contours renforcés indiquent le contrôleur de la relation).*

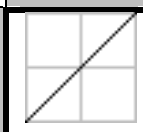
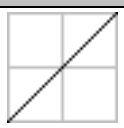
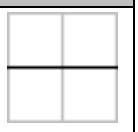
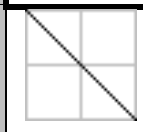
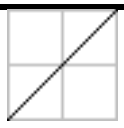
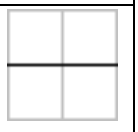
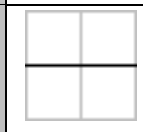
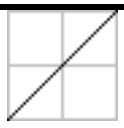
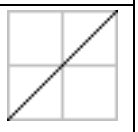
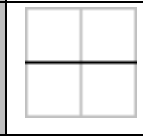
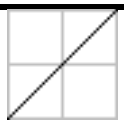
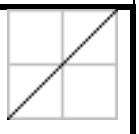
	A1	A2	A3
R1			
R21			
R22			
R3			

Tableau 5.29. *Les fonctions d'effet des relations sur les acteurs.*

Le tableau 5.30 présente les résultats des quatre cas de simulations. Dans les quatre cas, les deux acteurs A1 et A3 utilisent l'algorithme principal, tandis que A2 est anti-élitiste, protectionniste, égalitariste ou utilise l'algorithme principal. Les paramètres psycho-cognitifs et psycho-social sont initialisés de façon standard : discernement = 1, ténacité = réactivité = 5, égocentrisme = 0,5, et une répartition de 40% pour A2 et 50% pour A1 et A3 sur la dernière règle.

		A1	A2	A3	acteurMax	acteurMin	Satisfaction Globale
A2-Anti-élitiste	Satisfaction	11,72	**91,31**	**93,55**	94,45	8,24	196,58
A2-Protectionniste		63,97	58,97	90,23	90,37	**58,3**	213,17
A2-Égalitariste		56,92	58,51	67,16	**67,71**	**55,79**	182,59
A2-Algorithme principal		5	97	100	100	2	202
Satisfaction Globale Maximale		100	40	100	100	40	**240**

Tableau 5.30. *Satisfaction moyenne des acteurs à l'issue de 200 simulations d'un modèle à trois acteurs et quatre relations (A1 et A3 utilisent toujours l'algorithme principal, tandis que A2 est anti-élitiste, protectionniste, égalitariste ou utilise l'algorithme principal). La dernière ligne indique la satisfaction des acteurs dans la configuration correspondant au maximum de la satisfaction globale.*

Dans le cas où les trois acteurs utilisent l'algorithme principal, A1 et A3 dépend chacun à 50% de la relation qu'il contrôle, et ne peut donc que coopérer avec A2. Ce dernier coopère avec A1 dans 5% des simulations, bien qu'il n'ait aucun avantage à le faire.

Dans le cas où A2 est anti-élitiste, il augmente sa propre satisfaction s'il est le plus satisfait, et dans le cas contraire, diminue celle de l'acteur le plus satisfait, A3. Les satisfactions des deux acteurs sont très proches bien que A3 soit en moyenne légèrement plus satisfait que A2. En effet, A3 pose 5 points d'enjeux sur R22 tandis que A2 ne pose que 1 point, ce qui fait qu'une faible action positive sur cette relation augmente la satisfaction de A3 cinq fois plus que celle de A2. Dans le cas où A2 est protectionniste, il augmente la satisfaction de l'acteur le moins satisfait, A1, ce qui résulte en une « bonne » satisfaction pour A1 et A2, et sans impact particulier sur la satisfaction de A3, proche de son optimal. Dans le dernier cas où A2 est égalitariste, il parvient à ajuster l'état des deux relations qu'il contrôle de façon à moyenner les satisfactions des trois acteurs et donc à réduire l'écart entre les trois acteurs.

Il apparaît donc que la rationalité anti-élitisme n'est pas effective lorsqu'elle qui se ramène à de la jalousie, ce qui n'est pas le cas lorsqu'elle est associée au protectionniste. En effet, la structure du jeu fait que lorsque A2 participe à l'augmentation de la satisfaction de A1, cela diminue largement sa propre satisfaction (il pose 3 points d'enjeux sur la relation R21, et sa fonction d'effet sur R21 est l'inverse que celle de A1). A3 devient alors l'acteur toujours le plus satisfait sur lequel l'anti-élitisme de A2 s'applique de façon systématique, et donc efficace.

Un modèle à trois acteurs et quatre relations / anti-égalitariste

Le modèle d'organisation traité dans cette section diffère de celui étudié dans la section précédente par les enjeux des acteurs sur les relations et les fonctions d'effet, comme illustré dans les tableaux 5.31 et 5.32. Ce modèle a été construit pour mettre en évidence la différence entre les résultats de l'anti-égalitarisme d'une part, et de l'élitisme et l'anti-protectionnisme d'autre part. Il présente un conflit structurel entre A2 et A3 sur la relation R22 contrôlée par A2, et une similarité d'intérêt entre A1 et A2 compte tenu de leurs fonctions d'effet sur les relations R21 et R1. L'acteur A3 est nettement le plus influent, avec 16 points d'enjeux sur la relation R3 qu'il contrôle, mais c'est A2 qui est en situation intermédiaire entre A1 et A3. Si A2 est élitiste, on s'attend à ce qu'il augmente sa propre satisfaction quand il est le plus satisfait. Dans le cas contraire, il doit sacrifier

l'impact de la relation R22 qu'il contrôle en faveur de A3. S'il est anti-protectionniste, on s'attend à ce qu'il sacrifie l'impact de la relation qu'il contrôle afin de diminuer la satisfaction de A1. Anti-égalitariste, A2 devrait agrandir l'écart entre la satisfaction de l'acteurMax, lui-même ou A3, et l'acteurMin, A1.

	A1	A2	A3
R1	**3**	1	0
R21	4	**1**	0
R22	0	**1**	4
R3	3	7	**6**

Tableau 5.31. *Les enjeux des acteurs sur les relations (les contours renforcés indiquent le contrôleur de la relation).*

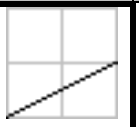
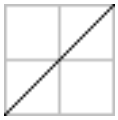
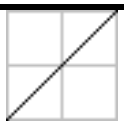
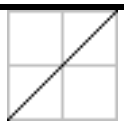
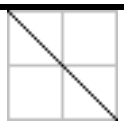
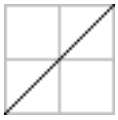
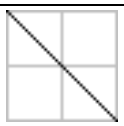
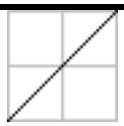
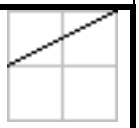
	A1	A2	A3
R1			
R21			
R22			
R3			

Tableau 5.32. *Les fonctions d'effet des relations sur les acteurs.*

Le tableau 5.33 présente les résultats des quatre cas de simulations, où les deux acteurs A1 et A3 utilisent l'algorithme principal, tandis que A2 est élitiste, anti-protectionniste, anti-égalitariste ou bien utilise l'algorithme principal. Les paramètres psycho-cognitifs et psycho-social sont initialisés de façon standard et une répartition de 30% pour A1, 20% pour A2, et 60% pour A3 sur la dernière règle.

		A1	A2	A3	acteurMax	acteurMin	Satisfaction Globale
A2-Élitiste	Satisfaction	9,25	87,6	67,6	**100**	9,25	164,45
A2-Anti-protectionniste		-61,58	75,85	44,77	85,62	**-61,58**	59,04
A2-Anti-Égalitariste		-13,18	85,62	50,27	**95,23**	**-13,83**	122,71
A2-Algorithme principal		9,85	91,5	53,6	100	9,85	154,95
Satisfaction Globale Maximale		10	80	100	100	80	**190**

Tableau 5.33. *Satisfaction moyenne des acteurs à l'issue de 200 simulations d'un modèle à trois acteurs et quatre relations (A1 et A3 utilisent toujours l'algorithme principal, tandis que A2 est élitiste, anti-protectionniste, anti-égalitariste ou utilise l'algorithme principal). La dernière ligne indique la satisfaction des acteurs dans la configuration correspondant au maximum de la satisfaction globale.*

Dans le cas où les trois acteurs utilisent l'algorithme principal, A3 dépend à 60% de la relation qu'il contrôle et ne peut que coopérer avec A2. Ce dernier coopère avec A3 dans 42% des simulations.

Lorsque A2 est élitiste, il augmente sa propre satisfaction quand il perçoit qu'il est le plus satisfait, et augmente celle de l'acteur A3 dans l'autre cas, sans impact particulier sur A1. Dans le cas où A2 est anti-protectionniste, c'est toujours A1 qui est l'acteur le moins satisfait avec une satisfaction proche de la pire configuration. Dans le cas où A2 est anti-égalitariste, il parvient à agrandir l'écart entre la satisfaction de l'acteurMax (A2 ou A3) et l'acteurMin (A1).

Contrairement à l'anti-élitisme dans la section précédente, il apparaît que, dans ce modèle d'organisation, la rationalité anti-protectionnisme appliquée seule est effective. En effet, la structure du jeu fait que A1 est toujours l'acteur le moins satisfait ; la participation de A2 à la détérioration de la situation de A1 est pénalisante pour A2, mais pas au point de le placer en situation d'acteur le moins satisfait.

L'analyse de sensibilité de l'égocentrisme de A2 (cf. 5.5.4) montre que plus la valeur de ce paramètre diminue, plus s'agrandit l'écart entre les satisfactions de l'acteurMax et de l'acteurMin.

5.5.3 – Analyse de sensibilité du paramètre égocentrisme / égalitariste

Nous reprenons le modèle considéré dans la section 5.5.2 sur la rationalité égalitariste. Les figures 5.26 et 5.27 montrent les résultats d'une analyse de sensibilité, comprenant 10 expériences où l'égocentrisme de A2 varie entre 0,1 et 1.

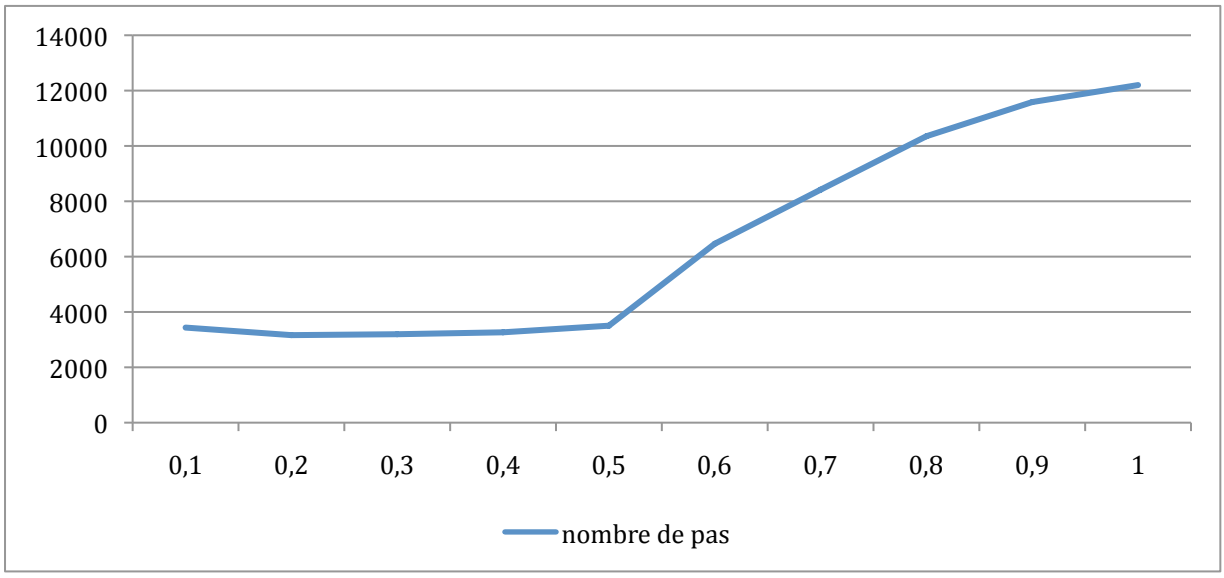


Figure 5.26. *Le nombre de pas moyen pour les simulations, en fonction de l'égocentrisme du A2 (A1 et A3 utilisent l'algorithme principal, tandis que A2 est égalitariste).*

La figure 5.27 montre que l'égalitarisme de A2 est inopérant pour un égocentrisme entre 0,7 et 1 et qu'il progresse entre 0,7 et 0,5, valeur en deçà de laquelle il est effectif. L'augmentation de la satisfaction de l'acteurMin (cf. figure 5.27) résultant de la diminution de l'égocentrisme de A2 fait que A1 est plus rapidement satisfait, ce qui réduit d'autant la durée des simulations (cf. figure 5.26).

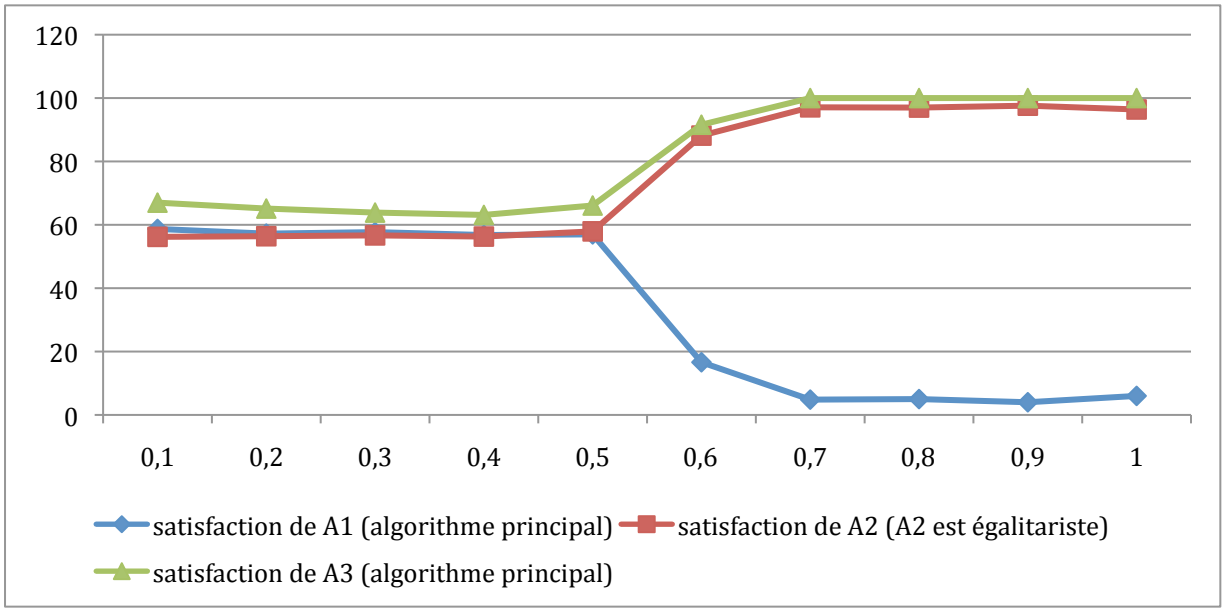


Figure 5.27. *La moyenne de la satisfaction de chaque acteur en fonction de l'égocentrisme du A2 (A1 et A3 utilisent l'algorithme principal, tandis que A2 est égalitariste).*

5.5.4 – Analyse de sensibilité du paramètre égocentrisme / anti-égalitariste

Dans cette section, nous étudions le modèle considéré dans la section 5.5.2 sur la rationalité anti-égalitariste. Les figures 5.28, 5.29 et 5.30 montrent les résultats d'une analyse de sensibilité, comprenant 10 expériences où l'égocentrisme de l'acteur A2 varie entre 0,1 et 1.

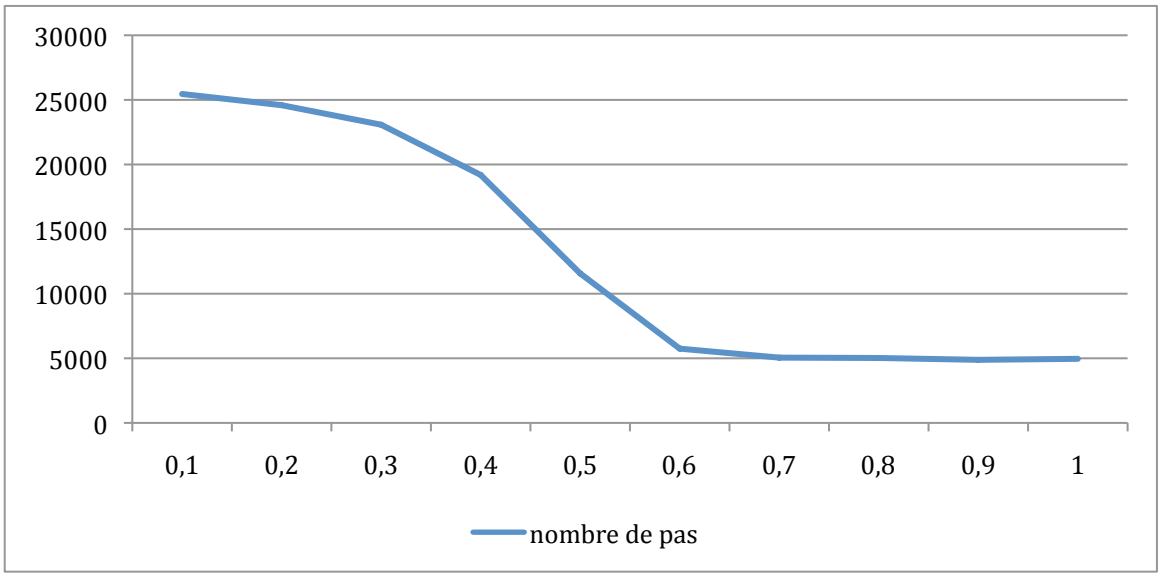


Figure 5.28. *Le nombre de pas moyen pour les simulations, en fonction de l'égocentrisme du A2 (A1 et A3 utilisent l'algorithme principal, tandis que A2 est égalitariste).*

La figure 5.29 montre que plus l'égocentrisme de A2 diminue, plus il s'acharne à l'augmentation de l'écart entre la satisfaction de l'acteurMin et celle de l'acteurMax. On observe le même phénomène que celui dans la figure 5.27 : La diminution de la satisfaction de l'acteurMin (cf. figure 5.29) résultant de la diminution de l'égocentrisme de A2 fait que A1 est plus difficilement satisfait, ce qui augmente d'autant la durée des simulations (cf. figure 5.28).

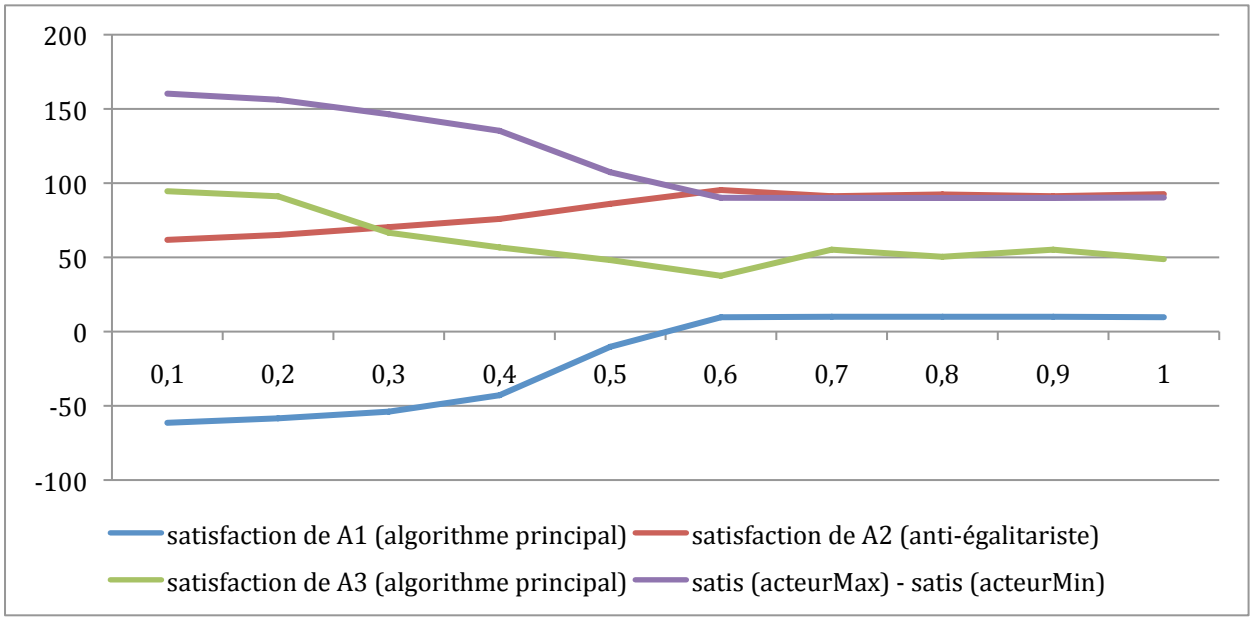


Figure 5.29. *La moyenne de la satisfaction de chaque acteur et celle de l'écart entre la satisfaction de l'acteurMax et celle de l'acteurMin, en fonction de l'égocentrisme de A2 (A1 et A3 utilisent l'algorithme principal, tandis que A2 est anti-égalitariste).*

Pour un égocentrisme entre 1 et 0,6, l'anti-égalitarisme de A2 n'a aucun effet, et que cet effet est de plus en plus marqué entre 0,6 et 0,1 (cf. figure 5.29). La satisfaction de A1 diminue continûment puisque la structure du jeu fait que nul ne peut lui contester la position d'acteur défavorisé. En faisant diminuer la satisfaction de A1, A2 diminue sa propre satisfaction – son anti-protectionnisme l'emporte sur son élitisme – au point d'être supplanté par A3 dans la position d'élite. En effet, sur l'intervalle 0,6 – 0,3, l'élitisme de A2 fait progresser la satisfaction de A3. La figure 5.30 montre la variation de l'état des deux relations contrôlées par A2, R21 appliquant l'anti-protectionnisme et R22 appliquant l'élitisme. Par contre l'état des deux autres relations (R1 et R3) reste quasiment constant à 10 quelque soit l'égocentrisme de A2.

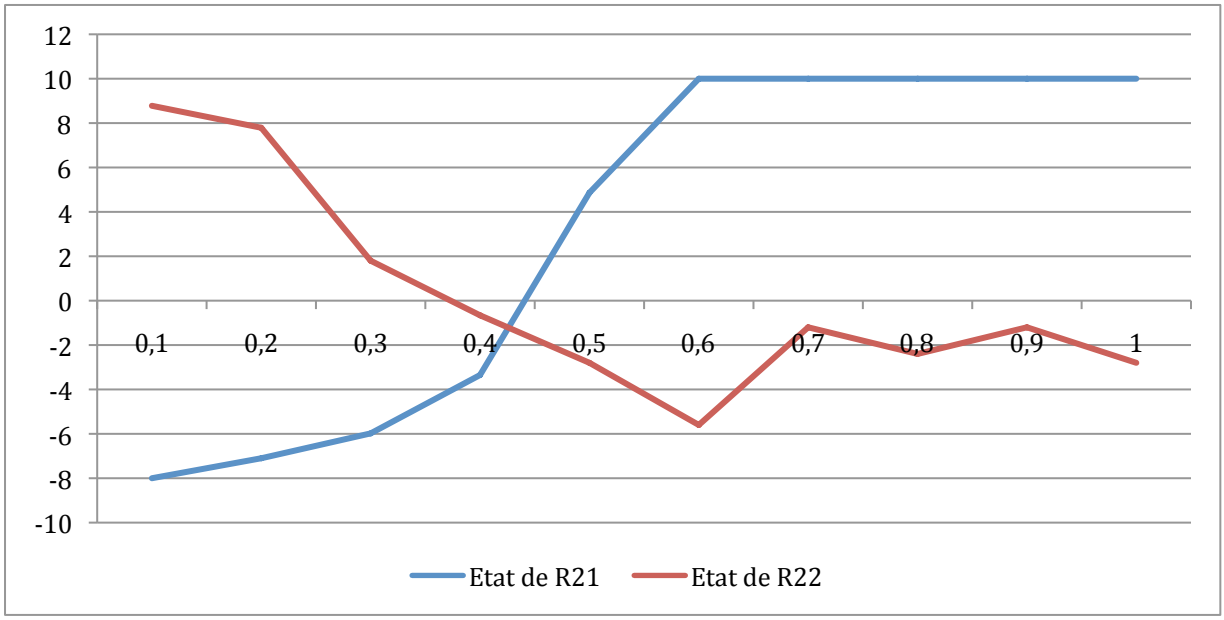


Figure 5.30. *La moyenne de l'état des deux relations R21 et R22, en fonction de l'égocentrisme de A2 (A1 et A3 utilisent l'algorithme principal, tandis que A2 est égalitariste).*

5.5.5 – Discussion

La rationalité de l'égalitariste associe celles de l'anti-élitiste et du protectionniste, et si un acteur qui exerce un contrôle sur la satisfaction des acteurs acteurMax et acteurMin adopte un comportement égalitariste, il va effectivement diminuer l'écart entre la satisfaction des acteurs. Cette diminution d'écart n'est pas toujours sans effet sur celui qui la provoque, puisque sa propre satisfaction peut s'en trouver diminuée (cf. figure 5.27). Son égocentrisme définit donc jusqu'à quel point son égalitarisme le conduira à accepter la diminution de sa propre satisfaction.

Il en est de même pour la rationalité de l'anti-égalitariste qui associe l'élitisme et l'anti-protectionnisme et qui cherche à augmenter l'écart entre les satisfactions des acteurs. Cette augmentation n'est pas toujours en faveur de celui qui la provoque et nécessite dans certains cas le sacrifice par cet acteur de sa propre satisfaction. C'est ce qui se produit dans le cas examiné en 5.5.4, où la maximisation de l'écart n'est possible que par la diminution de la satisfaction de l'acteur anti-égalitariste (cf. figure 5.29). Cette diminution n'est pas toujours acceptée par l'acteur anti-égalitariste, en fonction de la valeur de son égocentrisme.

Chapitre 6 – SocLab

SocLab, pour Social Laboratory, ou Laboratoire Social, est un environnement d'édition, d'analyse, et de simulation multi-agents dédié à l'expérimentation virtuelle d'organisations sociales. Ecrit en Java sous la licence GPL, il a fait l'objet d'un développement spécifique. Ce choix s'explique par la nécessité pour les différents membres du projet de disposer d'un outil complet, simple d'utilisation et d'installation, et dont l'apprentissage ne soit pas surchargé par des pré-requis autres que ceux de la SAO, plutôt qu'une interface entre différents logiciels tiers plus complexes à utiliser pour des non-informaticiens. D'autre part, la construction de SocLab s'est inscrite dans la démarche interdisciplinaire du projet, et devait répondre au fur er à mesure des différents besoins apparus au cours de la collaboration (visualisation, analyse de sensibilité...), d'où la nécessité d'un développement ad hoc.

Le logiciel SocLab est téléchargeable gratuitement depuis le site officiel du projet SocLab[24] où l'utilisateur peut également trouver une description de la SAO, du projet SocLab, des exemples d'organisations, et un tutoriel détaillé de l'utilisation de SocLab. Un fichier ReadMe a été aussi ajouté avec le logiciel pour faciliter la prise en main et le démarrage.

L'interface de SocLab est décomposée en plusieurs modules dont chacun a un objectif précis (par exemple : le module d'édition d'un modèle d'organisation). Pour chaque module, l'utilisateur peut consulter une aide, sous la forme d'un fichier pdf en français et en anglais, décrivant les différents éléments du module.

Dans ce chapitre, nous décrivons l'ensemble des différents modules de SocLab. Tout d'abord, nous présentons le module d'édition (§6.1) permettant au modélisateur de créer et d'éditer un modèle d'organisation. Puis nous présentons deux modules d'analyse des organisations : le premier (§6.2) permet d'analyser des états remarquables d'une organisation sous la forme de tableaux et de graphes, tandis que le deuxième (§6.3) permet d'analyser la structure d'une organisation sous la forme de réseaux. Ensuite nous présentons le module de simulation (§6.4) permettant de simuler le comportement des acteurs sociaux, ainsi que le module d'analyse de sensibilité des paramètres (§6.5). Finalement, nous présentons le module de visualisation des résultats de simulations et des analyses de sensibilité (§6.6). Pour terminer, nous présentons le module de génération des rapports (§6.7).

6.1 – Édition

Dans un premier temps, SocLab permet à l'utilisateur de définir la structure d'une organisation en créant les deux éléments principaux qui la constituent : les acteurs, et les relations (cf. figure 6.1). Au cours de cette édition, l'utilisateur peut décrire le modèle d'organisation, les acteurs, et les relations (notamment les interprétations des états) sous la forme d'un texte de description, pour documenter les choix de modélisation. Cela permet de donner du sens aux valeurs choisies, en étayant les choix par des explications de nature sociologique et surtout de laisser une trace de la justification des choix effectués.

[24] http://soclabproject.wordpress.com/

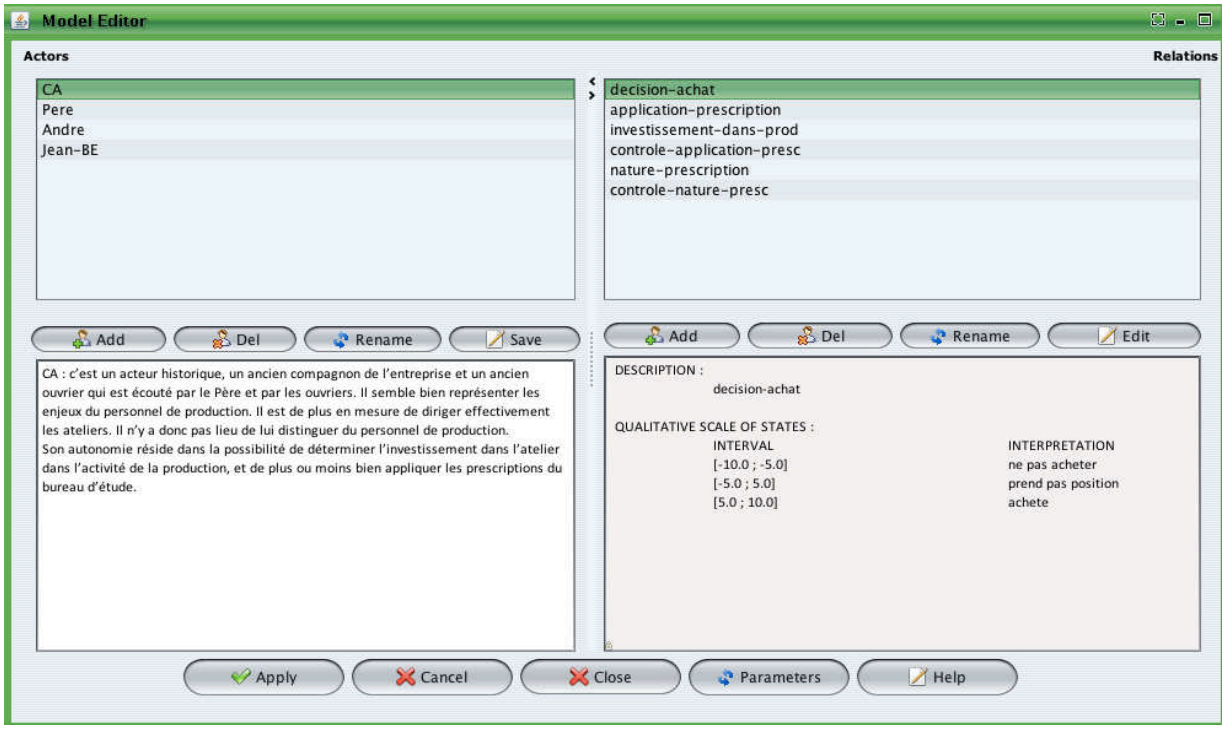
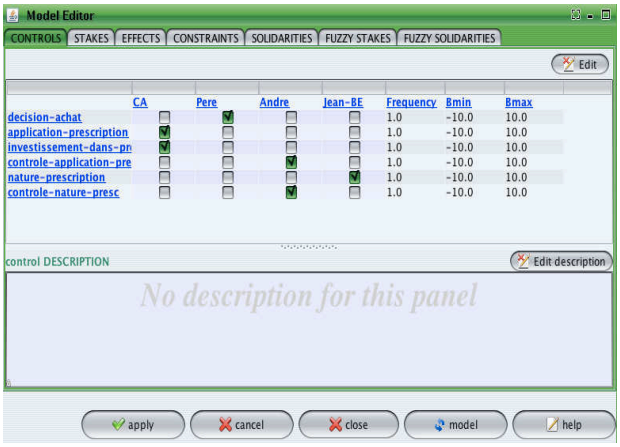


Figure 6.1. *Le module d'édition : acteurs et relations.*

Après avoir créer la liste des acteurs et des relations, l'utilisateur peut éditer la structure de cette organisation en modifiants les autres éléments qui la constituent : définir le contrôleur, la fréquence, et les bornes inférieures et supérieures de chacune des relations (cf. figure 6.2), distribuer les enjeux de chaque acteur sur les relations dont il dépend (cf. figure 6.3), définir les fonctions d'effets des relations sur les acteurs (cf. figure 6.4), définir les fonctions de contraintes entre les relations (cf. figure 6.5), préciser les solidarités entre les acteurs (cf. figure 6.6), et donner les ensembles valeurs minimales et maximales pour les enjeux ou les solidarités dans le cas de l'imprécision sur ces éléments (cf. figure 6.7).



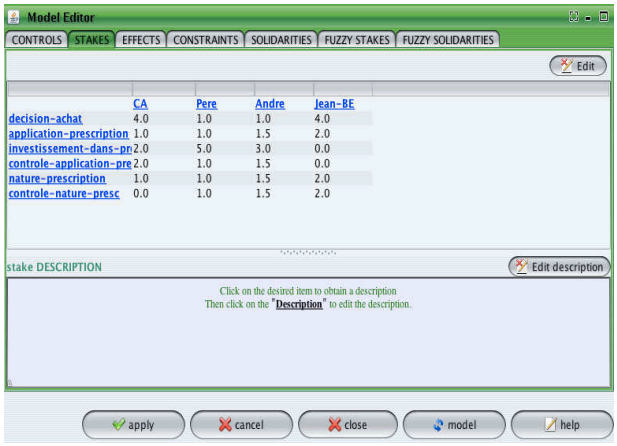


Figure 6.2. *Le module d'édition : le contrôle des relations.* **Figure 6.3.** *Le module d'édition : les enjeux des acteurs.*

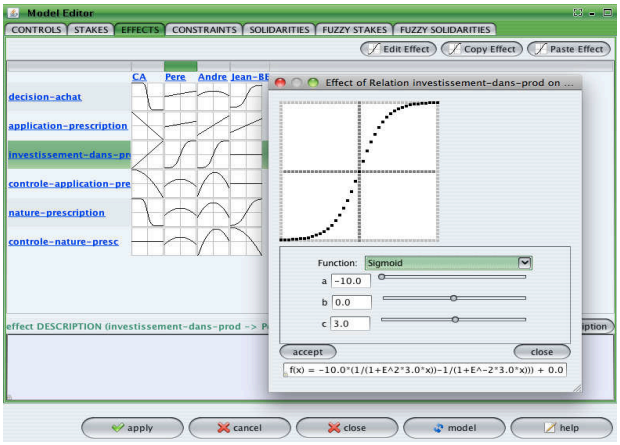


Figure 6.4. *Le module d'édition : les fonctions d'effet.*

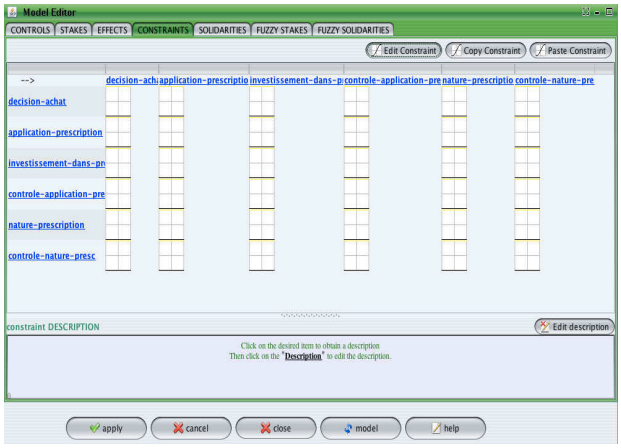


Figure 6.5. *Le module d'édition : les fonctions de contraintes.*

Figure 6.6. *Le module d'édition : les solidarités.*

Figure 6.7. *Le module d'édition : les enjeux imprécis.*

6.2 – Analyse d'états

Après avoir modélisé une organisation, le module d'analyse de SocLab permet à l'utilisateur d'explorer l'espace des états de l'organisation pour identifier différentes configurations et mettre en évidence certaines propriétés du modèle de l'organisation. Pour ce faire, l'utilisateur a la possibilité de créer et de modifier des configurations en précisant la valeur de l'état de chacune des relations. SocLab calcule et affiche alors les impacts reçus soit du fait des relations dont il dépend, soit du fait de ses solidarités, et les combine pour obtenir les valeurs de capacité d'action (ou de satisfaction) et de pouvoir (ou d'influence) correspondantes. L'utilisateur peut ainsi visualiser ces résultats sous la forme de tableaux (cf. figure 6.8).

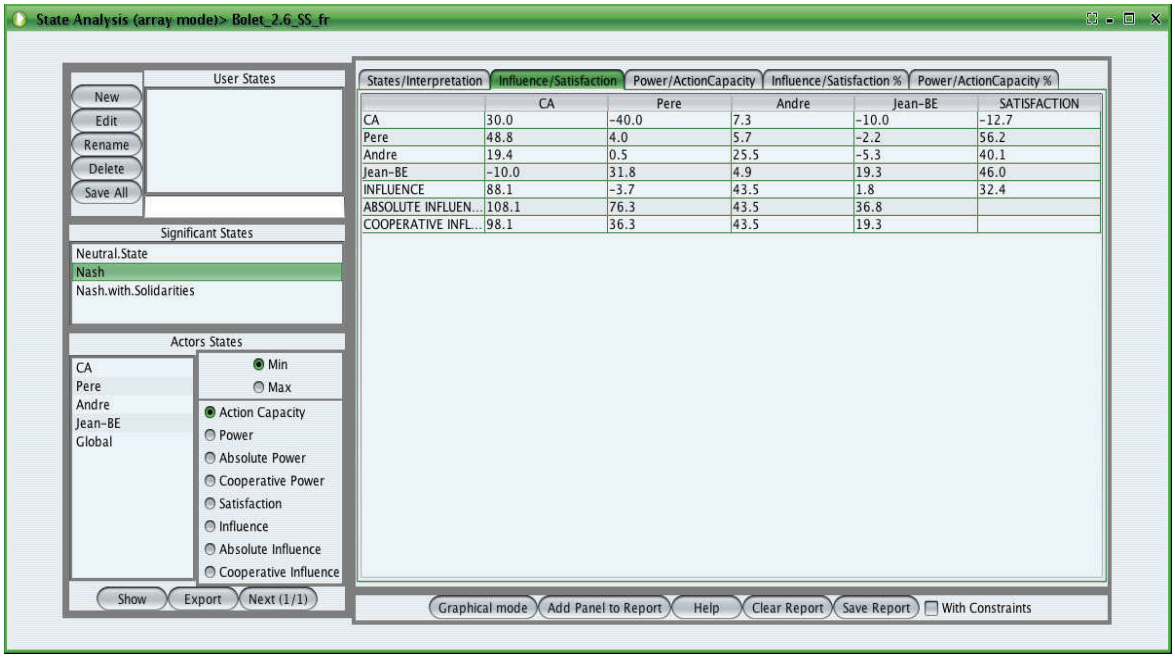


Figure 6.8. *Le module d'analyse des états remarquables d'une organisation, sous la forme des tableaux.*

Ainsi, le tableau à droite de la figure 6.8 se lit de la façon suivante, pour une configurations donnée (Nash) : l'acteur CA (en ligne) reçoit comme effet des relations contrôlées par lui même et par les acteurs Père, André, et Jean des effets de 30, -40, 7,3, et -10. Il en résulte une satisfaction de -12,7. En colonne, l'acteur CA octroie à lui même et aux autres acteurs des impacts de 30, 48,8, 19,4 et -10. Il en résulte une influence de 88.2. Ces valeurs sont aussi fournies en pourcentage, exprimé par rapport au maximum possible que la valeur peut prendre, ce qui en facilite l'interprétation. Ces résultats peuvent aussi être représentées sous forme d'histogrammes, représentant les niveaux de satisfactions des acteurs. Les parts dues aux relations contrôlées par les différents acteurs sont identifiées par des couleurs propres à chaque acteur et l'état des différentes relations est fixé par l'intermédiaire de curseurs (cf. figure 6.9).

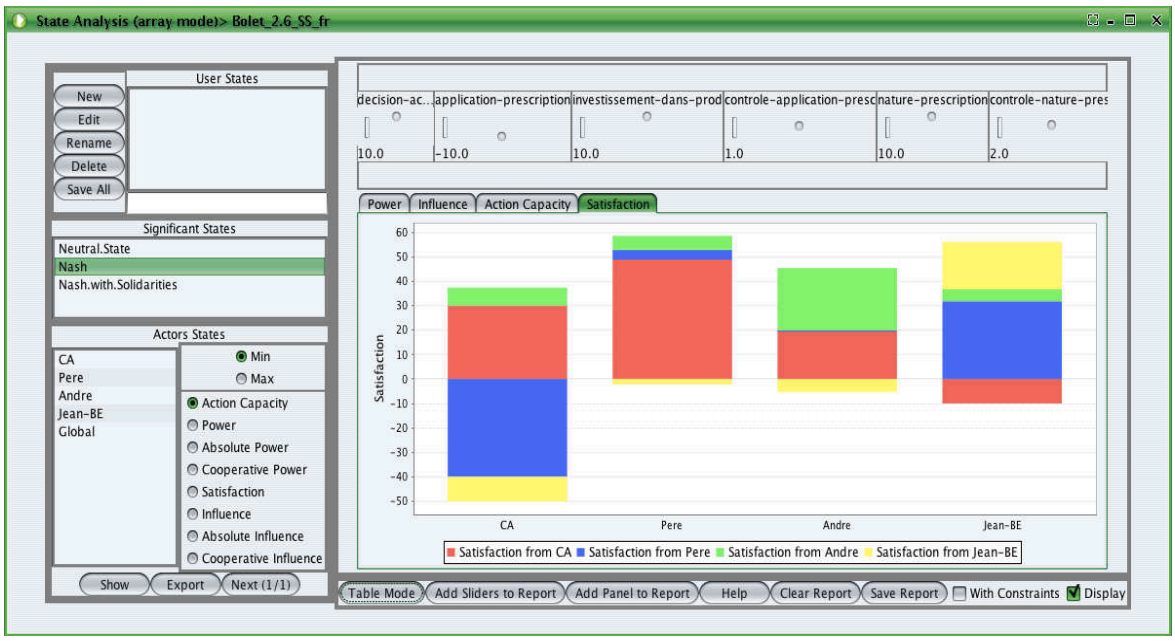


Figure 6.9. *Le module d'analyse des états remarquables sous la forme des graphes.*

Dans ce même module, il est également possible de parcourir l'espace des configurations de l'organisation pour rechercher celles qui maximisent ou minimisent certains critères, ou qui reflètent une configuration particulière du système (comme les équilibres de Nash).

Pour un acteur et un critère choisis par l'utilisateur, SocLab calcule les états qui maximisent ou minimisent ce critère, et offre la possibilité de faire défiler l'ensemble de ces états s'il y en a plusieurs. L'ensemble des critères disponibles comprend la capacité d'action, la satisfaction, le pouvoir et l'influence[25]. Il est aussi possible de calculer l'état qui maximise un critère de façon globale, c'est à dire en prenant en compte non plus la valeur du critère pour un acteur en particulier, mais la somme des critères pour tous les acteurs.

6.3 – Analyse structurelle

SocLab permet de représenter la structure d'un modèle d'organisation sous la forme d'un réseau d'acteurs, ou d'un réseau bipartite constitué de nœuds acteur et de nœuds relations (cf. figure 6.10). L'utilisateur peut sélectionner dans une liste les indicateurs qu'il souhaite voir étiqueter les nœuds ou les arêtes du graphe. Les indicateurs proposés sont les indicateurs structurels définis plus bas, plus représentatifs de ce qui est possible dans l'organisation et adaptés à ce genre de représentation de la structure d'une organisation :

- Strength : la force d'une relation sur un acteur. Elle détermine l'importance d'une relation sur la satisfaction d'un acteur. Il s'agit de mesurer l'amplitude maximale dans laquelle (l'état de) la relation peut affecter la satisfaction de l'acteur.
- Structural_Power : le pouvoir structurel d'un acteur sur un autre acteur. Il s'agit de mesurer la capacité potentielle d'un acteur à influer la satisfaction d'un autre acteur.
- Autonomy : l'autonomie d'un acteur. Elle est définie par la somme des enjeux que l'acteur pose sur les relations qu'il contrôle, normalisée par le total d'enjeux.
- Max_Satisfaction : la satisfaction maximale d'un acteur. Il s'agit de la valeur maximale que peut prendre la satisfaction d'un acteur.
- Min_Satisfaction : la satisfaction minimale d'un acteur. Il s'agit de la valeur minimale que peut prendre la satisfaction d'un acteur.
- Max_Potential_strength : la force potentielle maximale d'une relation. Il s'agit de la valeur maximale que peut prendre la force potentielle d'une relation sur les acteurs qui en dépendent. La force potentielle d'une relation r sur un acteur a est l'amplitude de la fonction d'effet de r sur a : Max (effet (r, a)) – Min (effet(r, a)). Elle diffère de la force d'une relation sur un acteur du fait qu'elle ne prend pas en compte l'enjeu que l'acteur pose sur cette relation.
- Accumulated_Potential_Strength : la force potentielle cumulée d'une relation. Il s'agit de la somme des forces potentielles d'une relation sur les acteurs qui en dépendent.
- Effective_Max_Strength : la force effective maximale d'une relation. Il s'agit de la valeur maximale que peut prendre la force d'une relation sur les acteurs qui en dépendent.
- Accumulated_Effective_Strength : la force effective cumulée d'une relation. Il s'agit de la somme des forces d'une relation sur les acteurs qui en dépendent.
- Relevance : la pertinence d'une relation. Il s'agit d'une évaluation de l'importance de cette relation compte tenu des enjeux que les acteurs placent dessus. C'est la somme pour chaque acteur A de l'enjeu qu'il pose sur cette relation, pondérée par la somme des

[25] Les variantes « Absolute » et « Cooperative » des indicateurs de pouvoir et d'influence sont les sommes des valeurs absolues, respectivement des valeurs positives, des termes qui composent ces indicateurs

solidarités (en valeur absolue) que chaque acteur B accorde à A.

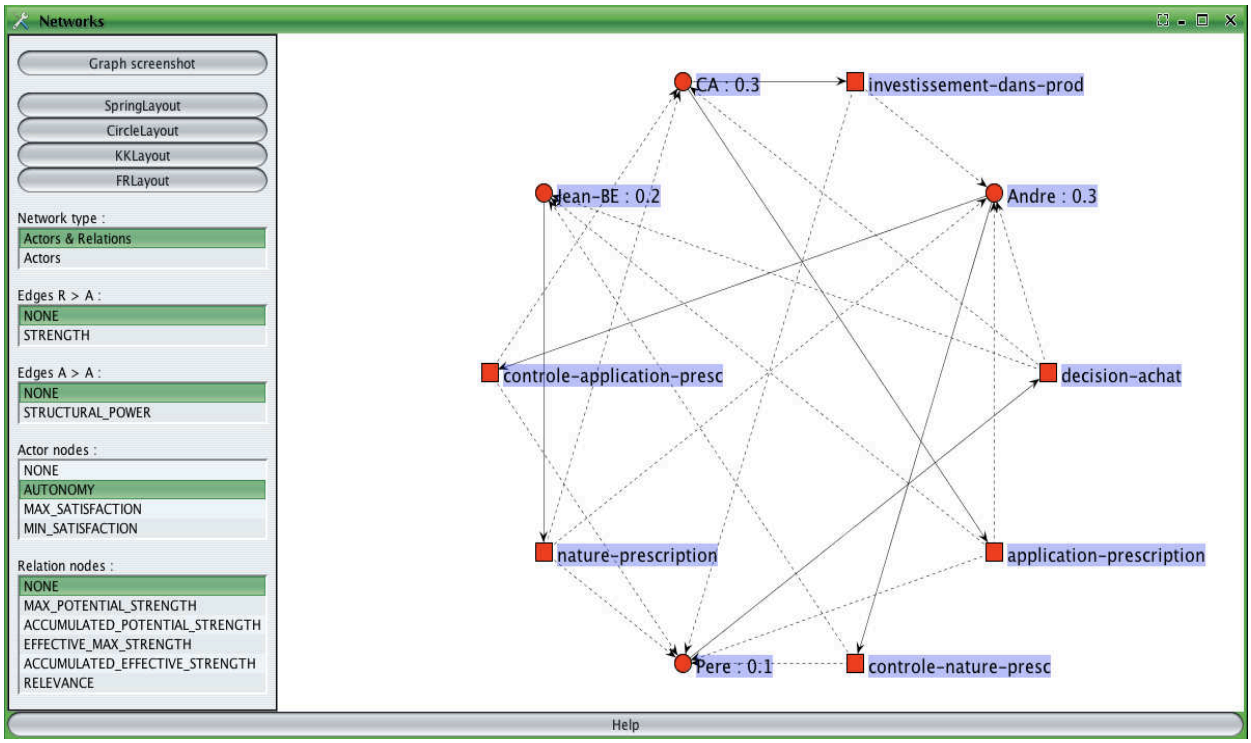
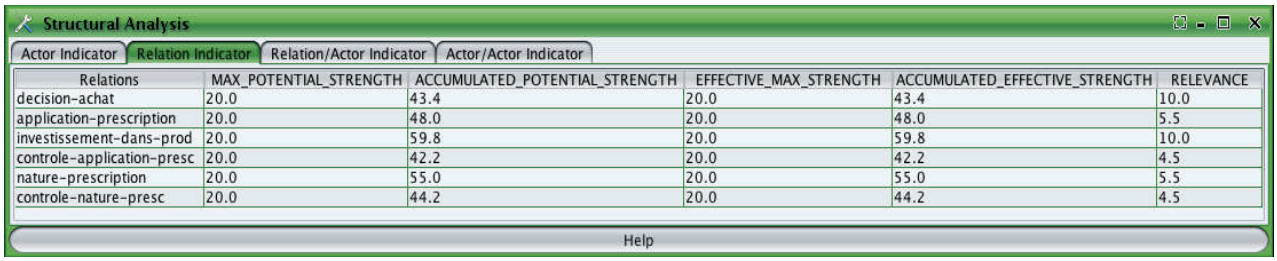


Figure 6.10. *Le module pour la représentation de la structure d'une organisation sous la forme de réseaux.*

Les valeurs des indicateurs peuvent être représentées sous la forme de tableaux (cf. figure 6.11) qui peuvent être copier depuis SocLab vers n'importe quel document textuel ou feuille de calcul, ce qui facilité l'utilisation des valeurs calculées par SocLab par n'importe quel autre outil.



Relations	MAX_POTENTIAL_STRENGTH	ACCUMULATED_POTENTIAL_STRENGTH	EFFECTIVE_MAX_STRENGTH	ACCUMULATED_EFFECTIVE_STRENGTH	RELEVANCE
decision-achat	20.0	43.4	20.0	43.4	10.0
application-prescription	20.0	48.0	20.0	48.0	5.5
investissement-dans-prod	20.0	59.8	20.0	59.8	10.0
controle-application-presc	20.0	42.2	20.0	42.2	4.5
nature-prescription	20.0	55.0	20.0	55.0	5.5
controle-nature-presc	20.0	44.2	20.0	44.2	4.5

Figure 6.11. *Le module d'analyse structurelle d'affichage d'indicateurs sous la forme de tableaux.*

6.4 – Simulation du comportement des acteurs sociaux

Le module de simulation, dont l'algorithme a été présenté dans les chapitres quatre et cinq, dispose d'une interface permettant d'initialiser l'état des relations, de préciser la valeur des paramètres psycho-cognitifs des acteurs, et de choisir la rationalité de chaque acteur (ParetoOptimum, Élitiste, Nash, …) (cf. figure 6.12). Il est possible d'associer une description à chaque *expérience de simulation* afin d'en garder une trace explicative. Il est aussi possible d'utiliser le mode *flou*, pour les enjeux et les solidarités des acteurs, afin de tester la robustesse du modèle face à l'imprécision de ces données. L'interface permet de fixer le nombre de simulations à lancer et le nombre de pas maximal de chaque simulation.

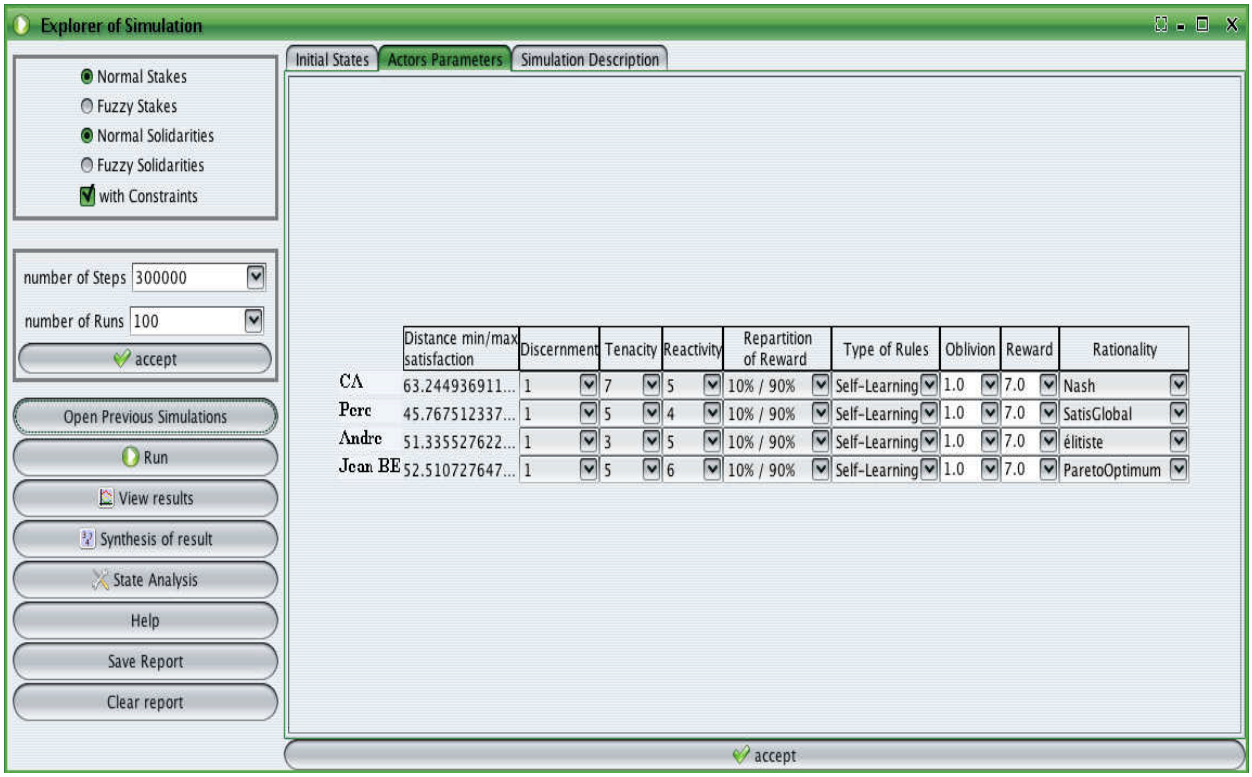


Figure 6.12. *Le module de simulation du comportement des acteurs. La première colonne (Distance min/max satisfaction) est la distance euclidienne entre la situation qui maximise la satisfaction de l'acteur et celle qui la minimise). Oblivion et Reward sont deux paramètres utilisés dans l'ancien type des règles présentées dans [Mailliard, 2008].*

Après avoir réalisé une expérience de simulation d'un modèle d'organisation, il est possible de représenter les résultats sous la forme de courbes montrant l'évolution de la valeur des états des relations au cours du temps (par exemple la courbe à gauche dans la figure 6.13 représente l'évolution de l'état de la relation « nature-prescription » dans le cas Bolet), ainsi que l'évolution de la satisfaction et l'ambition des acteurs (par exemple les deux courbes à droite dans la figure 6.13 représentent l'évolution de la satisfaction (en bleu) et de l'ambition (en rouge) de l'acteur CA dans le cas Bolet). Il est possible de visualiser soit les trajectoires d'un acteur ou d'une relation sur l'ensemble de toutes les simulations d'une expérience, soit les trajectoires de tous les acteurs ou toutes les relations au cours d'une simulation particulière. Il est aussi possible d'obtenir la satisfaction moyenne des acteurs et le pourcentage des simulations qui ont convergé (cf. figure 6.14), et d'autres paramètres comme l'état moyen des relations, le nombre de pas moyen des simulations qui ont convergé, etc.

Il reste à noter que dans les analyses des résultats, nous distinguons entre les simulations qui ont convergé et celles qui n'ont pas convergé et considérées inutiles du fait que, à la fin de ces simulations, un état d'équilibre n'a pas été trouvé par les acteurs. Les résultats affichés ne concernent donc que les simulations qui ont convergé.

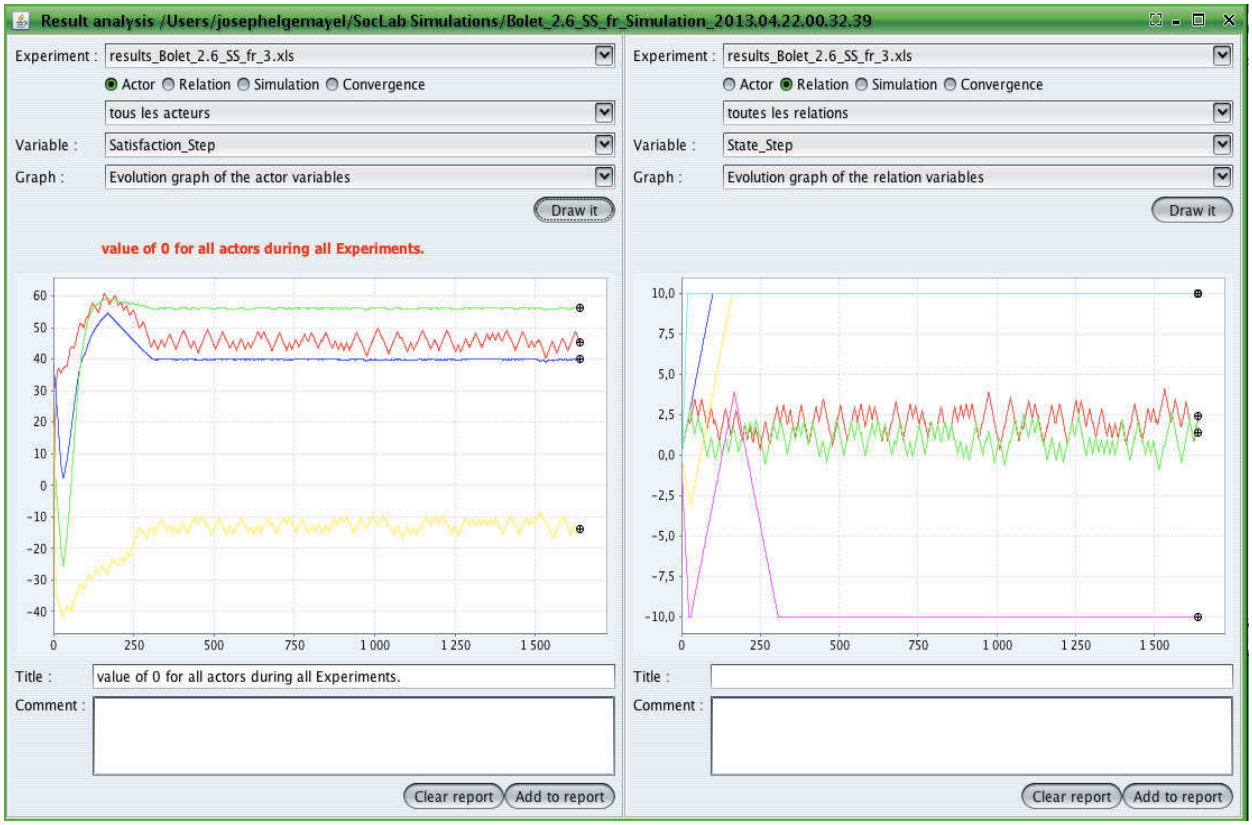


Figure 6.13. *Le module d'affichage des résultats d'une simulation du cas Bolet où tous les acteurs utilisent l'algorithme Nash, sous la forme de la trajectoire des acteurs (à gauche) ou des relations (à droite).*

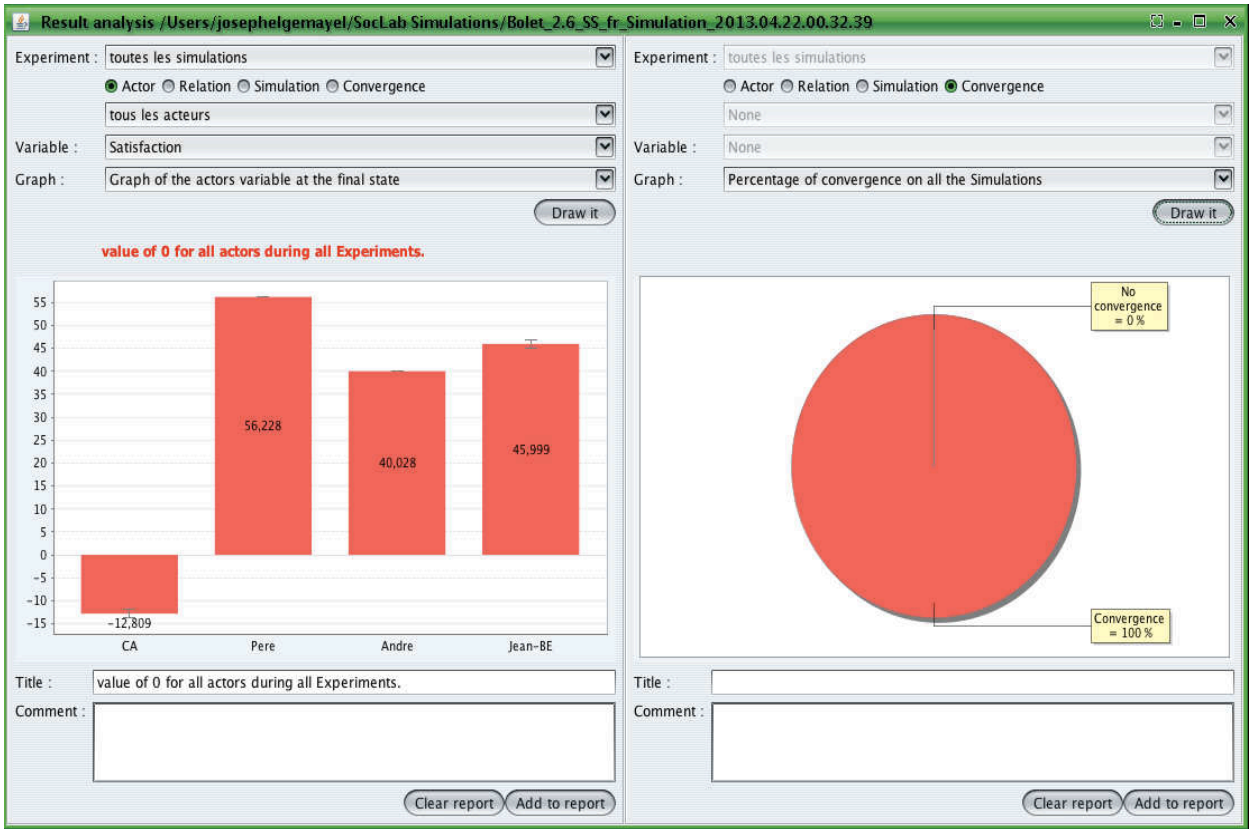
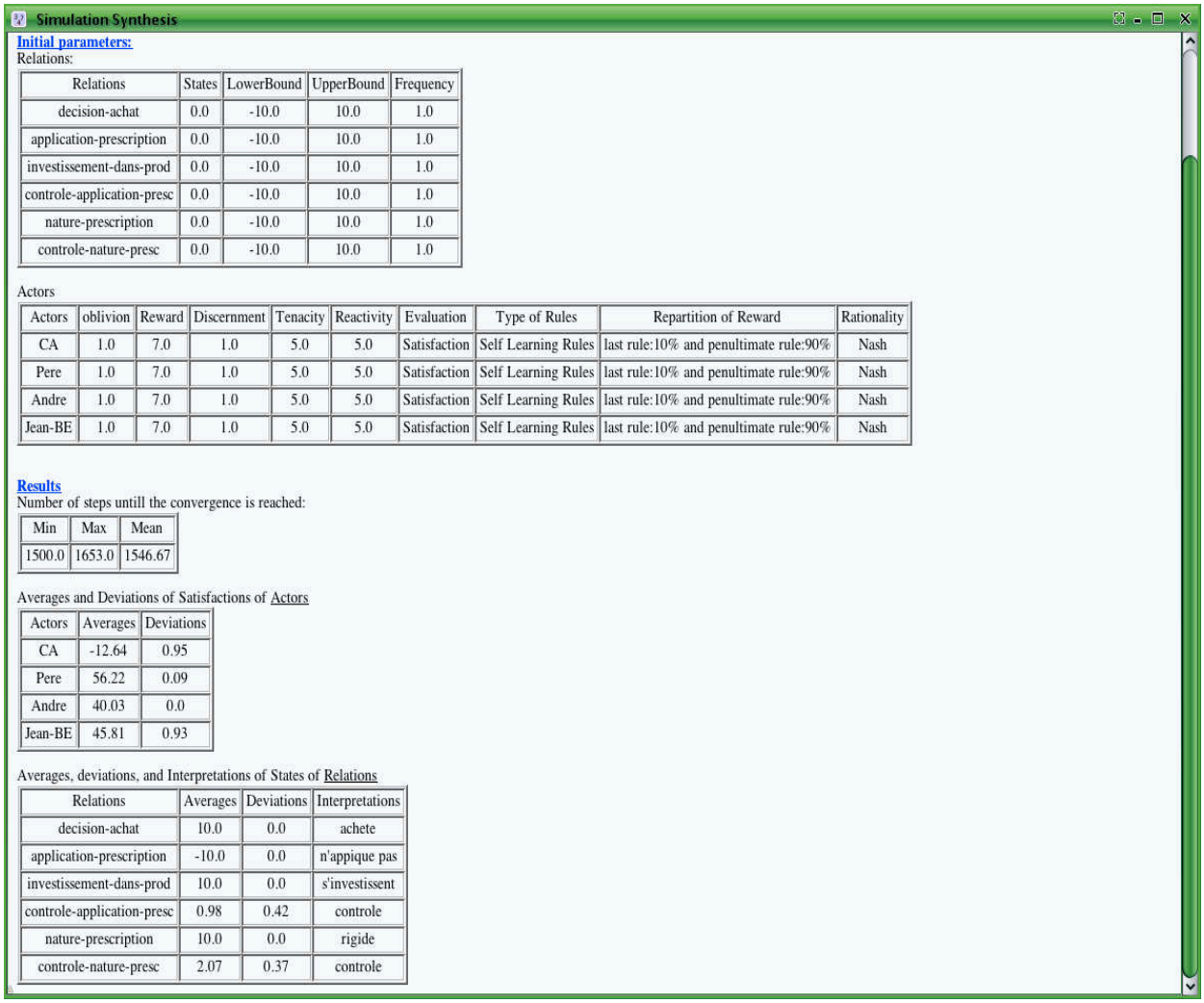


Figure 6.14. *Le module d'affichage des résultats de simulations du cas Bolet où tous les acteurs utilisent l'algorithme Nash, sous la forme d'histogrammes des satisfactions des acteurs (à gauche) et de secteurs de pourcentage de convergence (à droite).*

Il est aussi possible d'obtenir une synthèse des résultats de simulations sous la forme de tableaux. Tout d'abord, les valeurs initiales de l'état des relations et celles des paramètres des acteurs sont présentées. Ensuite, le nombre de pas nécessaire pour la convergence (minimal, maximale et moyen), et les moyennes des valeurs finales de l'état des relations et des satisfactions des acteurs sont indiqués avec les écarts types (cf. figure 6.15).


Simulation Synthesis

Initial parameters:

Relations:

Relations	States	LowerBound	UpperBound	Frequency
decision-achat	0.0	-10.0	10.0	1.0
application-prescription	0.0	-10.0	10.0	1.0
investissement-dans-prod	0.0	-10.0	10.0	1.0
controle-application-presc	0.0	-10.0	10.0	1.0
nature-prescription	0.0	-10.0	10.0	1.0
controle-nature-presc	0.0	-10.0	10.0	1.0

Actors

Actors	oblivion	Reward	Discernment	Tenacity	Reactivity	Evaluation	Type of Rules	Repartition of Reward	Rationality
CA	1.0	7.0	1.0	5.0	5.0	Satisfaction	Self Learning Rules	last rule:10% and penultimate rule:90%	Nash
Pere	1.0	7.0	1.0	5.0	5.0	Satisfaction	Self Learning Rules	last rule:10% and penultimate rule:90%	Nash
Andre	1.0	7.0	1.0	5.0	5.0	Satisfaction	Self Learning Rules	last rule:10% and penultimate rule:90%	Nash
Jean-BE	1.0	7.0	1.0	5.0	5.0	Satisfaction	Self Learning Rules	last rule:10% and penultimate rule:90%	Nash

Results

Number of steps untill the convergence is reached:

Min	Max	Mean
1500.0	1653.0	1546.67

Averages and Deviations of Satisfactions of Actors

Actors	Averages	Deviations
CA	-12.64	0.95
Pere	56.22	0.09
Andre	40.03	0.0
Jean-BE	45.81	0.93

Averages, deviations, and Interpretations of States of Relations

Relations	Averages	Deviations	Interpretations
decision-achat	10.0	0.0	achete
application-prescription	-10.0	0.0	n'appique pas
investissement-dans-prod	10.0	0.0	s'investissent
controle-application-presc	0.98	0.42	controle
nature-prescription	10.0	0.0	rigide
controle-nature-presc	2.07	0.37	controle

Figure 6.15. *Le module d'affichage des résultats de simulations du cas Bolet où tous les acteurs utilisent l'algorithme Nash, sous la forme de tableaux.*

6.5 – Analyse de sensibilité

Le module d'analyse de sensibilité (cf. figure 6.16) complète le module de simulation pour permettre d'étudier l'influence de certains paramètres sur le déroulement des simulations. A ce stade du développement de SocLab, il est seulement possible de faire varier les paramètres de l'algorithme de rationalité des acteurs, ainsi que certains éléments du modèle de l'organisation comme les solidarités ou les enjeux des acteurs. L'utilisateur choisit le ou les paramètres dont il veut étudier l'influence et l'intervalle à l'intérieur duquel il désire les faire varier. Ce module se charge alors de lancer une série d'expériences de simulations au cours desquelles la valeur du paramètre choisi est tirée aléatoirement dans cet intervalle.

Les résultats sont présentés sous formes de courbes, de la même façon que pour les résultats de simulation.

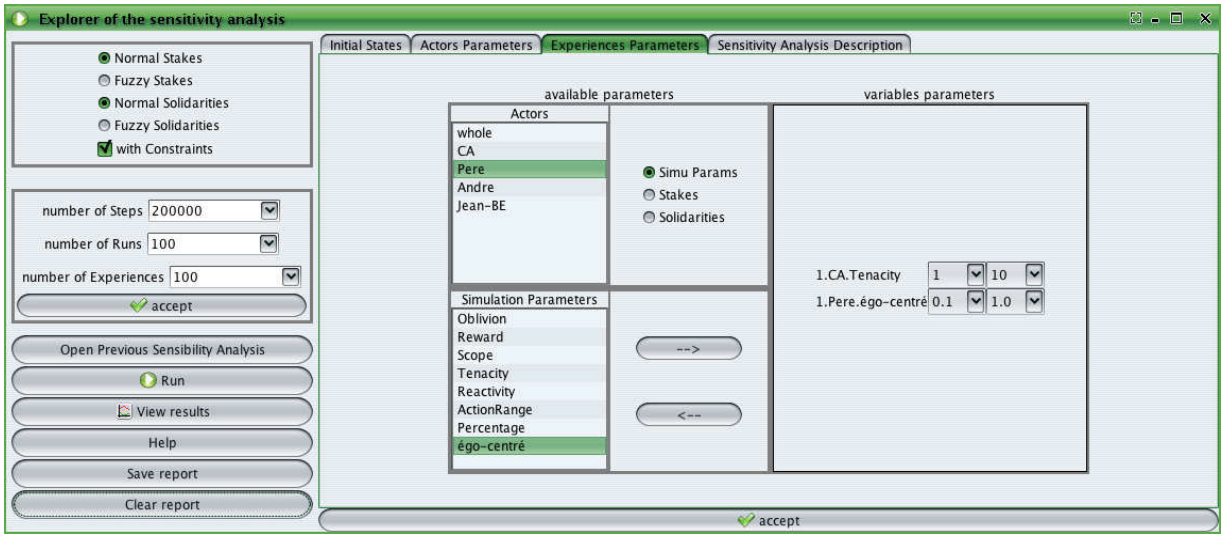


Figure 6.16. *Le module d'analyse de sensibilité.*

6.6 – Génération de rapports

Dans chacun des modules de SocLab, l'utilisateur a la possibilité de générer, sous forme d'un fichier au format .RTF, .cvs ou .png, différents rapports lui permettant de garder la trace de son travail. Cette fonctionnalité est essentielle car on est amené à produire un certain nombre de modèles d'une organisation concrète donnée avant de parvenir au modèle qui rend compte de façon satisfaisante du problème que l'on veut étudier. Il est alors essentiel de pouvoir comparer ces modèles, leurs propriétés et résultats de simulations, et les raisons qui ont motivé leur élaboration. Le rapport généré par le module d'édition contient toutes les informations nécessaires pour reproduire le modèle : acteurs, relations, enjeux, solidarités et forme des fonctions d'effets, avec l'éventuelle justification de chacune des valeurs. Le module d'analyse des configurations génère des rapports contenant les tableaux ou graphiques de l'analyse des configurations sélectionnés par l'utilisateur. Enfin, les modules de simulation et d'analyse de sensibilité permettent de générer un rapport contenant un résumé des résultats des simulations, ainsi que les courbes choisies par l'utilisateur.

6.7 – Format des fichiers

Indépendamment des rapports générés à la demande de l'utilisateur, SocLab utilise trois types de format de fichiers.

Tout d'abord, un modèle d'organisation est sauvegardé sur le disque dur sous la forme d'un fichier texte (avec l'extension .org) au format XML. Dans ce fichier, l'utilisateur peut retrouver facilement les différents éléments de la structure de cette organisation : les acteurs et leurs paramètres, les relations et leurs états, et les liens entre eux : enjeux, fonctions d'effet, solidarités, etc. La sauvegarde au format XML permet à l'utilisateur d'éditer une organisation de façon facile et efficace même en dehors de SocLab.

Par ailleurs, les résultats d'une expérience de simulation sont rassemblés dans un répertoire dont le nom est composé de trois parties sous la forme suivante : NomOrga_`Simulation`_DateHeure (par exemple `Concerto_Simulation_2012.07.13.22.12.27`). La partie NomOrga est le nom du fichier .org contenant le modèle de l'organisation, la partie Simulation permet de distinguer les répertoires des simulations et les répertoires des analyses de sensibilité, et la partie DateHeure est l'heure de lancement de l'expérience.

Ce répertoire contient :

- Un fichier, dont le nom est de la forme NomOrga_`SimulationInitialParameters`.xls (exemple : `Concerto_SimulationInitialParameters.xls`), enregistrant la valeur de tous les paramètres de l'expérience (cf. figure 6.12): le nombre de simulations, le nombre d'étapes maximal de chaque simulation, les valeurs initiales de l'état des relations, et les valeurs des paramètres psycho-cognitifs des acteurs.
- n fichiers contenant des informations détaillés pour chaque simulation, dont le nom est de la forme `results`_NomOrga.run.txt, où run est le numéro de la simulation, allant de 0 à n-1, et NomOrga est le nom du modèle de l'organisation. Chaque fichier contient :
    - La liste des noms des acteurs et relations du modèle.
    - S'il y a eu convergence, et le nombre de pas nécessaire.
    - La satisfaction finale et l'ambition finale de chaque acteur.
    - L'état final de chaque relation.
    - L'évolution, à chaque pas de la simulation, de la satisfaction et de l'ambition de chaque acteur.
    - L'évolution, à chaque pas de la simulation, de l'état de chaque relation.

  La structure de ce fichier permet en fait de déterminer des variables (1) résultats de la simulation (e.g. le nombre de pas) (2) associées à chaque acteur ou à chaque relation en fin de simulation et (3) associées à chaque acteur ou à chaque relation à chaque pas de la simulation, qui seront prises en charge par le module d'affichage des résultats comme montré par les figures 6.13 et 6.14.
- Un fichier `synthesis`.NomOrga.txt qui contient les données pour le traitement statistique des résultats de l'expérience. Il s'agit d'un tableau avec une ligne pour chaque simulation et en colonne :
    - La durée, en nombre de pas, de la simulation.
    - La valeur finale de la satisfaction de chaque acteur.
    - La valeur finale de l'ambition de chaque acteur.
    - La valeur finale de l'état de chaque relation.

Les résultats d'une analyse de sensibilité sont rassemblés dans un répertoire, dont le nom est composé de trois parties : NomOrga_`Sensitivity`_DateHeure (exemple : `Concerto_Sensitivity_2012.07.13.22.12.27`) où les parties NomOrga et DateHeure sont définies comme pour les simulations.

Ce répertoire contient :

- Un fichier, dont le nom est de la forme NomOrga_`SensitivityInitialParameters`.xls (exemple : `Bolet_SensitivityInitialParameters.xls`), permettant de retrouver les paramètres de cette analyse : le nombre d'expériences de simulation, le nombre de simulations pour chaque expérience, le nombre de pas maximal pour chaque simulation, les valeurs initiales des états des relations, les valeurs des paramètres psycho-cognitives des acteurs, et la liste des paramètres à analyser avec leurs valeurs minimales et maximales.
- Un répertoire pour chacune des expériences de l'analyse, dont le nom est le numéro de cette expérience et le contenu comme indiqué ci-dessus pour les expériences de simulation.
- Un fichier pour chacune des n expériences, dont le nom est de la forme `param_exp`run`.txt`, contenant les informations sur l'initialisation de chaque expérience, run allant de 0 à n-1.

- Un fichier pour chacune des n expériences, dont le nom est de la forme `results_exp`run.txt, contenant les informations concernant les résultats de l'expérience (satisfactions finales des acteurs, états finaux des relations, et le nombre de pas en moyenne), run allant de 0 à n-1.
- Un fichier « `sensitivity_result_`NomOrga`.xls` » qui résume les résultats de l'analyse de sensibilité et permettra de faire des traitements statistiques.

Chapitre 7 – Conclusion

7.1 – Les résultats

Dans ce mémoire, nous avons étudié plusieurs modèles d'organisations permettant de mettre en évidence les différentes propriétés des algorithmes présentés dans les chapitres quatre et cinq. Ces modèles sont des instances du méta-modèle des organisations sociales présenté dans le chapitre deux, dont l'originalité et l'intérêt est d'être sociologiquement fondée sur une théorie sociologique : la Sociologie de l'Action Organisée.

L'algorithme principal présenté dans le chapitre 4 modélise comment les acteurs d'une organisation, et plus généralement d'un système d'action collective, peuvent ajuster durablement leurs comportements les uns aux autres, en négociant leurs gestions des relations dont ils contrôlent l'accès en échange de la gestion de celles dont ils ont besoin pour atteindre leurs objectifs. Conformément à la sociologie de l'action organisée, le comportement des acteurs est stratégique : chacun cherche à préserver ou à améliorer sa satisfaction, ce qui permet l'émergence d'une régulation du système, propriété constitutive des organisations. Les acteurs n'ont qu'une rationalité limitée, en raison de l'opacité des relations sociales et de leurs limitations cognitives ; de ce fait, ils cherchent à trouver une situation satisfaisante qui n'est pas nécessairement un optimum. Enfin, ce comportement est globalement coopératif ; en effet, cette coopération, lorsque la structure du SAC la permet, est indispensable à son bon fonctionnement que chacun des acteurs a intérêt à maintenir.

Les résultats de cet algorithme sont dans l'ensemble proches des optima de Pareto, conformément à ce que l'on en attend. Globalement, plus la structure de l'organisation valorise la coopération des acteurs, *i.e.* plus l'écart entre le minimum et le maximum de la satisfaction globale est important, plus les acteurs coopèrent et plus les simulations convergent rapidement. Le principe de l'apprentissage sur lequel il est fondé permet de ne pas préjuger de la façon dont les acteurs sociaux déterminent leurs comportements et il est en accord avec les observations empiriques : dans tout système d'action, il y a une période d'adaptation, pendant laquelle chacun observe les réactions des autres à son propre comportement, avant que les comportements se stabilisent. Il est vraisemblable du point de vue cognitif : il demande peu de ressources et de compétences, les acteurs étant essentiellement réactifs. Il est aussi vraisemblable du point de vue social dont il respecte l'opacité : un acteur a besoin d'une évaluation de ce à quoi il peut prétendre (pour l'initialisation de son ambition) et d'une connaissance purement locale de son propre état – à savoir une évaluation de l'écart entre son ambition et sa satisfaction courante et de la distance entre sa situation courante et celles des règles contenues dans sa base de règles – mais il n'a besoin d'aucune connaissance sur la structure du jeu. Enfin, même si la notion de solidarité entre les acteurs permet de représenter des intérêts collectifs, cet algorithme n'échappe pas aux critiques que l'on peut formuler à l'encontre de l'utilitarisme [Boudon, 2007]. Notons cependant que la rationalité des acteurs est bien *procédurale* : l'ensemble des alternatives est cherché (dans les règles existantes) ou construit (par la création d'une nouvelle règle) contextuellement à chaque étape et le critère d'utilité d'un acteur, à savoir son ambition, est lui aussi déterminé contextuellement.

Quant à la pertinence des paramètres psycho-cognitifs de cet algorithme, la ténacité et la réactivité d'un acteur sont les deux paramètres psycho-cognitifs les plus importants. La ténacité détermine la propension de l'acteur à privilégier l'exploration ou l'exploitation, tandis que la réactivité détermine l'importance relative que l'acteur accorde au présent et au passé dans son processus d'apprentissage. Ces deux paramètres influencent les résultats de simulations (simulation plus ou moins longue et coopération plus ou moins élevée) et permettent de prendre en compte les caractéristiques individuelles des acteurs sociaux. En ce qui concerne les deux autres paramètres (le discernement et la répartition du renforcement), la valeur qu'il convient de donner à ces deux paramètres est fortement liée à la structure de l'organisation. En effet, le discernement détermine la

capacité d'un acteur à distinguer entre les différentes situations selon leur plus ou moins grande proximité, et son influence sur les résultats de simulation dépend donc de la présence ou non de situations effectivement différentes. Tandis que la répartition du renforcement détermine, durant la mise à jour de la qualité des règles, la proportion de l'impact direct (à l'étape t+1) d'une règle sur la satisfaction de l'acteur et celle de l'impact différé (à l'étape t+2). Ce procédé de mise à jour permet à un acteur de prendre en compte la réaction des autres à ses actions, et donc de coopérer dans les cas où la coopération est nécessaire pour d'améliorer la satisfaction des autres. La valeur qui convient pour ce paramètre dépend donc de la difficulté à détecter la coopération des autres acteurs.

L'extension de cet algorithme présentée à la fin du chapitre quatre, pour les contextes où les acteurs ont plusieurs objectifs incomparables, hérite de tous les avantages de l'algorithme principal. En effet, cette extension est basée sur le même principe et permet aux acteurs de poursuivre simultanément plusieurs objectifs hétérogènes. Ces objectifs sont représentés par des relations incommensurables où chaque acteur cherche à maximiser l'impact de chaque relation indépendamment des autres relations. Les résultats de cet algorithme sont dans l'ensemble satisfaisants. Cependant, il ne faut pas s'attendre à ce que l'acteur arrive toujours à améliorer sa satisfaction quand il utilise cet algorithme. Par exemple dans le modèle free-rider (cf. 4.6.2), l'algorithme multi-critère permet à A1 de mieux analyser sa situation et de détecter l'acteur qui ne coopère pas. Cependant, il n'est pas en mesure de le pénaliser. En effet, A1 contrôle une seule relation dont les trois autres acteurs dépendent, et ses actions affectent directement les trois acteurs de la même façon ; une mauvaise action lui coute très cher et en résulte par la non-coopération des trois acteurs en même temps.

Nous avons aussi proposé la définition de modèles rationalités des acteurs qui conduiraient une organisation à se stabiliser vers d'autres configurations qui satisfont une propriété remarquable, tels que les équilibres de Nash, la maximisation de la satisfaction globale de l'organisation, etc. Les résultats de ces algorithmes sont dans l'ensemble conformes à ceux attendus, pour autant que la configuration la configuration soit « socialement faisable », c'est-à-dire qu'aucun acteur n'ait les moyens d'empêcher son instauration. Il ne faut donc pas s'attendre à ce que l'organisation se régule toujours dans une configuration proche de celle visée par les acteurs. Par exemple, dans le cas Seita où MainE est en position d'avantage structurel très marqué, munir MainE d'une rationalité protectionniste permet d'améliorer la satisfaction des acteurs les moins satisfaits, tandis que en munir uniquement les autres acteurs ne changea rien aux résultats. Globalement, plus les acteurs qui adoptent un type de rationalité spécifique (par exemple protectionniste) sont dotés d'un pouvoir leur permettant de diriger le jeu, plus les simulations convergent vers la configuration visée par les acteurs (par exemple, l'augmentation de la satisfaction de l'acteur le moins satisfait). Dans certains cas, la participation d'un acteur à la réalisation d'un objectif social autre que son propre intérêt le pénalise largement. Par exemple, dans le dilemme du prisonnier où l'acteur élitiste participe à l'augmentation de la satisfaction de l'acteurMax, cette augmentation n'est possible que par la détérioration de la situation de l'acteur élitiste. Pour cette raison, un paramètre psycho-social, l'égocentrisme d'un acteur, a été introduit pour déterminer jusqu'à quel point l'acteur accepte la détérioration de sa propre situation afin de réaliser son objectif social. Plus la valeur de ce paramètre diminue, plus l'acteur accepte de contribuer à la réalisation de son objectif secondaire et accepte la détérioration de sa propre situation, et par conséquent plus les simulations convergent vers la configuration visée par l'acteur.

Nous pensons que ces algorithmes ont une portée beaucoup plus générale que le cadre de la SAO. Le « jeu social » peut être appliqué dans des contextes tels que la commande et le contrôle, la gestion de crise, ou encore dans les modèles de coordination sociale pour les sociétés d'agents logiciels autonomes, et d'une façon générale, dans le cadre du contrôle distribué d'un système très partiellement observable.

7.2 – Les perspectives

Dans la version actuelle des algorithmes présentés dans les chapitres quatre et cinq de cette thèse, les valeurs des paramètres psycho-cognitifs et psycho-social sont fixées par le modélisateur selon les hypothèses qu'il veut tester au début de la simulation. On peut se poser la question inverse : une structure organisationnelle donnée étant fixée, quelles doivent être les caractéristiques comportementales des acteurs pour que cette organisation fonctionne convenablement ? Une perspective de recherche est donc de permettre à chaque acteur d'adapter la valeur de ces paramètres à sa situation, c'est-à-dire de modifier leur valeur au fur et à mesure du déroulement de la simulation. L'idée sous-jacente est que l'acteur comprend de mieux en mieux la structure de l'organisation et le contexte de son action [Sigaud *et al.*, 2003]. Par exemple, un acteur peut augmenter sa ténacité d'une simulation à une autre quand il s'aperçoit que cela pourrait améliorer sa satisfaction de façon raisonnable sans augmenter la durée des simulations. De même il peut faire diminuer son égocentrisme d'une simulation à l'autre quand il s'aperçoit qu'il peut participer davantage à la réalisation de son objectif social sans faire trop diminuer sa propre satisfaction. En outre, un acteur peut aussi faire diminuer sa réactivité durant la même simulation quand il se rend compte que sa situation n'est pas stable et que sa satisfaction varie amplement de façon stochastique et rapide, si bien la configuration de l'organisation ne s'achemine pas vers les conditions d'une convergence. De même, il peut faire croître son discernement durant la même simulation quand il rencontre plusieurs situations différentes (par exemple, sa satisfaction change de sens bien qu'aucun acteur n'a changé son comportement).

Par ailleurs, d'autres versions des algorithmes méritent d'être étudiées. On pourrait bien sûr utiliser des opérateurs plus sophistiqués que l'addition et la multiplication pour la mise à jour de la valeur des variables (cf. 4.2.2). On pourrait faire expliciter la négociation du comportement entre les acteurs, et inclure la réputation et la confiance où la confiance est le lubrifiant de la négociation et la réputation renforce la confiance. De même, on pourrait ajouter des émotions dans le comportement des acteurs. L'idée sous-jacente est que nos émotions peuvent accélérer et guider nos décisions, mais elles peuvent aussi les entraver et les fausser. Par exemple, un acteur animé d'une grande motivation, d'envie et d'impatience, va explorer de façon inexplicable et essayer toute action possible lui conduisant à une éventuelle augmentation de sa satisfaction. En revanche, un acteur animé du doute ou de la peur n'osera même pas à coopérer avec les autres acteurs afin d'éviter de se retrouver dans sa pire situation.

Bibliographie

Andreae, J. H. (1969). Learning machines--a unified view. In *Encyclopedia of Information, Linguistics, and Control*, Meetham, A. R., Hudson, R. A. (Eds.), p. 261-270. Pergamon, Oxford.

Axelrod, R. (1992). Donnant, donnant. Théorie du comportement coopératif. *Editions Odile Jacob*, 1992.

Bakker, B., Steingrover, M., Schouten, R., Nijhuis, E., Kester, L. (2005). Cooperative multi-agent reinforcement learning of traffic lights. Presented at the Workshop Coop. Multi-Agent Learn., 1*6th Eur. Conf. Mach. Learn. (ECML-05)*, Porto, Portugal, Oct. 3.

Bakker, P., Kuniyoshi, Y. (1996). Robot See, Robot Do : An Overview of Robot Imitation. In *AISB96 Workshop on Learning in Robots and Animals*.

Baldet, B. (2012). Gérer la rivière ou la crue ? Le gouvernement du risque d'inondation entre enjeux localisés et approche instrumentée. Le cas de la vallée du Touch en Haute-Garonne. PhD Dissertation in Sociology, *University of Toulouse*, June 26th 2012.

Barel Y. (1979). Le paradoxe et le système. *Essai sur l'imaginaire social*, Grenoble, PUG, 1979.

Barrat, S., Tabbone, S. (2010). Modélisation, classification et annotation d'images partiellement annotées avec un réseau Bayésien. *Actes du 17ème Congrès Francophone de Reconnaissance des Formes et Intelligence Artificielle (RFIA 2010)*, Cean, 2010.

Barto, A. G. (1985). Learning by statistical coopération of self-interested neuron-like computing éléments. *Human Neurobiology*, 4 : 229-256.

Barto, A. G., Sutton, R. S., Anderson, C. W. (1983). Neuronlike elements that can solve difficult learning control problems. *IEEE Trans. Syst.*, Man Cybern. 13 : 835-846.

Bellman, R. (1957). A Markovian Decision Process. *Indiana Univ. Math*. J. 6 No. 4, pp 679-684.

Bellman, R. (1957a). Dynamic Programming. *Princeton University Press*, Princeton, NJ.

Bernoux, P. (1985). La sociologie des organisations. *Seuil*.

Bertsekas, D. P., Tsitsiklis, J. N. (1996). Neural Dynamic Programming. *Athena Scientific*, Belmont, MA.

Bertsekas, D. P. (1995). Dynamic Programming and Optimal Control. *Athena, Belmont*, MA.

Bertsekas, D. P. (1989). Dynamic programming : Deterministic and stochastic models. *Englewood Cliffs, NJ : Prentice-Hall*.

Boudon R. (2007). Essais sur la théorie générale de la rationalité, *PUF*, 2007.

Boutilier, C. (1996). Planning, learning and coordination in multi-agent decision processes. In *Proc. 6th Conf. Theor. Aspects Rationality Knowl. (TARK-96)*, De Zeeuwse Stromen, The Netherlands, p. 195-210, March 1996.

Brown, G. W. (1951). Iterative solutions of games by fictitious play. In *Activity Analysis of Production and Allocation*, T.C. Koopmans, Ed. New York : Wiley, 1951, ch. XXIV, p. 374-376.

Busoniu, L., Babuska, R., De Schutter, B. (2008). A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Transactions on systems, man, and cybernetics – part C : Applications and Reviews*, vol. 38, no. 2.

Busoniu, L., De Schutter, B., Babuska, R. (2005). Multiagent reinforcement learning with adaptive state focus. In *proceedings 17th Belgian-Dutch Conference on Artificial Intelligence (BNAIC-05)*, Brussels, Belgium, Oct 17-18, p. 35-42.

Chapron, P. (2012). Modélisation et analyse des organisations sociales : propriétés structurelles, régulation des comportements et évolution. Thèse de doctorat, *Université des Sciences Sociales*, Toulouse, France, 2012.

Chelghoum, N., Zeitouni, K., Laugier, T., Fiandrino, A., Loubersac, L. (2006). Fouille de données spatiales. Approche basée sur la programmation logique inductive. *EGC2006* : 529-540.

Claus, C., Boutillier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems, In *Proc. 15th Nat. Conf. Artif. Intell. 10th Conf. Innov. Appl. Artif. Intell. (AAAI/IAAI-98)*, Madison, WI, Jul. 26-30, p. 746-752.

Clegg, S. R. (2003). Handbook of organization studies. *Sage Publication*, 2003. URL : http://www.worldcat.org/oclc/249400076

Clouse, J. (1995). Learning from an automated training agent. Presented at the *worshop agents that learn from other agents, 12th International Conference on Machine Learning (ICML-95)*, Tahoe City, CA, July 9-12.

Conitzer, V., Sandholm, T. (2003). AWESOME : A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. In *Proc. 20th Int. Conf. Mach. Learn. (ICML-03)*, Washington, DC, Aug. 21-24, p. 83-90.

Cosine, S. (2005). Comment l'empathie vient aux enfants. *La recherche, n° spécial Grandir, l'enfant et son développement 388, Sophia publications*, juillet-août 2005.

Craik, K. J. W. (1943). The Nature of Explanation. *Cambridge University Press*, Cambridge, UK.

Crites, R. H., Barto, A. G. (1998). Elevator group control using multiple reinforcement learning agents. *Mach. Learn.*, vol. 33, no. 2-3, p. 235-262, 1998.

Crozier M., Friedberg E. (1977). L'acteur et le système. *Contraintes de l'action collective*, Paris, Seuil, 1977.

Crozier, M. (1963). Le phénomène bureaucratique. *Essai sur les tendances bureaucratiques des systèmes d'organisation modernes et sur leurs relations en France avec le système social et culturel*, Paris, Seuil, 1963.

Dal Forno A., Merlone U. (2002). A multi-agent simulation platform for modeling perfectly rational and bounded-rational agents in organizations. *Journal of Artificial Societies and Social Simulation*, vol. 5, n° 2, march 2002, p. 433-438.

Dalmagro F., Jimenez J., Jimenez R., Lugo H. (2006). Bounded-rational-prisoners' dilemma : on critical phenomena of cooperation. *Applied Mathematics and Computation*, 176 (2), 2006, p. 462-469.

De Ketele, J. M, Chastrette, M., Cros, D., Mettelin, P., Thomas, J. (1988). Guide du formateur. *Collection Pédagogies en développement*. Bruxelles : De Boeck, 254 p.

Dean, T. L., Wellman, M. P. (1991). Planning and Control. *San Mateo, CA : Morgan Kaufmann*.

Delahaye, J. P. (1992). L'altruisme récompensé ? *Pour la Science (French Edition of Scientific American)*, 181 : 150-156, 1992.

Dennett, D. C. (1978). Why the law of effect will not go away. In *Brainstorms, by D. C. Dennett*, p. 71-89, Bradford Books, Montgomery, Vermont.

Dolk, V. (2010). Survey Reinforcement Learning.

Dreyfus, S. E., Law, A. M. (1977). The Art and Theory of Dynamic Programming. *Academic Press,* New York.

Dugatkin, L. A. (1997). Cooperation among Animals : An Evolutionary Perspective. *Oxford University Press*, 1997.

Durkheim, E. (1897). Le Suicide, Etude de Sociologie. *Livre Deuxième : Causes sociales et type sociaux*, chap. 5, II, 1897.

Dutech, A., Samuelides, M. (2003). Apprentissage par renforcement pour les processus décisionnels de Markov partiellement observés. *Revue d'Intelligence Artificielle, RIA*, vol. 17(4), 2003.

Edmonds B. (1999). Modelling bounded rationality in agent-based simulations using the evolution of mental models. *Computational Techniques for Modelling Learning in Economics, Kluwer Academic Publishers*, T. Brenner (Eds.), Boston, Massachussetts, 1999, p. 305-332.

El Gemayel, J., Sibertin-Blanc, C., Chapron, P. (2011). Impact of Tenacity upon the Behaviors of Social Actors. Dans : *Advances in Practical Multi-Agent Systems*, Quan Bai, Noaki Fukuta (Eds.), Springer, p. 287-306, Vol. 325, 2011.

ENSTA, Apprentissage par renforcement. http://wwwdfr.ensta.fr/Cours/docs/C10-2/chapitre7.pdf

Faris, E. (1926). The Concept of Imitation. *American Journal of Sociology, The University of Chicago Press*. Vol. 32, No. 3, pp. 367-378.

Fernandez, F., Parker, L. E. (2001). Learning in large cooperative multi-robot systems. *Int. J. Robot. Autom.*, vol. 16, no. 4, p. 217-226, 2001.

Fischer, F., Rovatsos, M., Weiss, G. (2004). Hierarchical reinforcement learning in communication-mediated multi-agent coordination. In *Proc. 3rd Int. Joint Conf. Auton. Agents Multi-Agent Syst. (AAMAS-04)*, New York, p. 1334-1335, August 2004.

Friedberg E. (1993). Le Pouvoir et la Règle. *Dynamiques de l'action organisée*, Paris, Seuil, 1993.

Friedberg, E. (1988). L'Analyse sociologique des organisations. *L'Harmattan*, Paris, nouv. Ed. Remise à jour édition.

Fromont, E., Quiniou, R., Cordier, M. O. (2006). Apprentissage multi-source par programmation logique inductive. *15e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle*.

Gabaix X., Laibson D. (2000). A Boundedly Rational Decision Algorithm. *AEA Papers and Proceedings*, vol 90, may 2000, p. 443-438.

Gigerenzer G., Goldstein D.G. (1996). Reasoning the fast and frugal way : Models of bounded rationality. *Psychological Review*, 103, 1996, p. 650-669.

Guestrin, C., Lagoudakis, M. G., Parr, R. (2002). Coordinated reinforcement learning. In *proceedings 19th International Conference on Machine Learning (ICML-02)*, Sydney, Australia, July 8-12, p. 227-234.

Hoffmann, R. (2000). Twenty Years on : The Evolution of Cooperation Revisited. *Journal of Artificial Societies and Social Simulation (JASSS)*, vol. 3, no. 2, 2000.

Ishiwaka, Y., Sato, T., Kakazu, Y. (2003). An approach to the pursuit problem on a heterogeneous multiagent system using reinforcement learning. *Robot. Auton. Syst.*, vol. 43, no. 4, p. 245-256, 2003.

Jehiel P. (1998). Learning to play limited forecast equilibria. *Games and Economic Behavior*, vol. 22, 1998, p. 274-298.

Kellerhals, J., Coenen-Huther, J., Modak, M. (1988). Figures de l'équité : La construction des normes de justice dans les groupes. *PUF-Le Sociologue*, Paris, 1988.

Kok, J. R., Hoen, P. J., Bakker, B., Vlassis, N. (2005). Utile coordination : Learning interdependencies among cooperative agents. In *Proc. IEEE Symp. Comput. Intell. Games (CIG-05)*, Colchester, U.K., Apr. 4-6, p. 29-36.

Kumar, V., Kanal, L. N. (1988). The CDP : A unifying formulation for heuristic search, dynamic programming, and branch-and-bound. In *Search in Artificial Intelligence*, Kanal, L. N., and Kumar, V. (Eds.), p. 1-37. Springer-Verlag.

Kumar, P. R., Varaiya, P. P. (1986). Stochastic Systems : Estimation, Identification, and Adaptive Control. *Prentice Hall, Englewood Cliffs*, New Jersey.

Kumar, P. R. (1985). A survey of some results in stochastic adaptive control. *SIAM J. Contr. Optim.* 23 : 329-380.

Le Hy, R., Arrigoni, A., Bessière, P., Lebeltel, O. (2004). Teaching Bayesian behaviours to video game characters. *Robotics and Autonomous Systems*, 47(2-3) : 177-185.

Littman, M. L. (2001). Value-function reinforcement learning in Markov games. *J. Cong. Syst. Res.*, vol. 2, no. 1, p. 55-66.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Proc. 11th Int. Conf. Mach. Learn. (ICML-94)*, New Brunswick, NJ, Jul. 10-13, pp. 157-163.

Lupi P. (1998). The Propagation of Cooperation in a Model of Learning with Endogenous Aspirations. *Research in Economics 98-06-052e*, Santa Fe Institute, June 1998. URL : http://ideas.repec.org/p/wop/safire/98-06-052e.html

Macy, M. W., Flache, A. (2002). Learning Dynamics in Social Dilemnas. *Proceedings of the National Academy of Sciences U.S.A*. May 14 ; 99(10) : 7229-36, 2002.

Mailliard, M., Sibertin-Blanc, C. (2010). What is Power ? Perspectives from Sociology, MAS and Social Networks. *Proceedings of the Social Networks Analysis and Norms for MAS Symposium (SNAMAS'10)*, G. Andrighetto, G. Boella, U. Pagallo, S. Villata (Eds.), Leicester (UK), 29/03 – 01/04/2010, p. 4-9, De Montfort University.

Mailliard, M. (2008). Formalisation Multi-Agents de la sociologie de l'Action Organisée. Thèse de doctorat, *Université des Sciences Sociales*, Toulouse, France.

March J. (1991). Exploration and exploitation in organizational Learning. *Organization Science,* vol. 2, n° 1, February 1991, p. 71-87.

Matignon, L., Mouaddib, A-I., Jenapierre, L. (2012). Coordinated Multi-Robot Exploration Under Communication Constraints Using Decentralized Markov Decision Processes. *International Conference on Advanced Artificial Intelligence (AAAI)*, 2012.

Metropolis, N., Stan, U. (1949). The Monte Carlo Method. *Journal of the American Statistical Association*. Offprint from Volume 44, Number 247.

Minsky, M. L. (1961). Steps towards artificial intelligence. *Proceedings of the Institute of Radio Engineers*, 49:8-30. Reprinted E. A .Feigenbaum, J. Feldman, editors, Computers and Thought. McGraw-Hill, New York, 406-450, 1963.

Minsky, M. L. (1954). Theory of neural-analog reinforcement systems and application to the brain-model problem. PhD thesis, *Princeton University*, Princeton, NJ.

Moga, S., Gaussier, P. (1998). Apprentissage par imitation. In *Neurosciences et Sciences pour l'Ingénieur (NSI98)*, Colmar.

Molm, L. D. (1991). Affect and Social Exchange : Satisfaction in Power-Dependence Relations. *American Sociological Review*, vol. 56, no. 4, p. 475-493, Aug., 1991.

Moore, A. W., Atkeson, C. G. (1993). Prioritized sweeping : Reinforcement learning with less data and less real time. *Machine Learning*, 13.

Morgenstern, O., Von Neumann, J. (1953). The Theory of Games and Economic Behavior, 3rd edition. *Princeton Universtiy Press*, 1953.

Nash, J. (1951). Non-Cooperative Games. *The Annals of Mathematics*, 2nd Ser., Vol. 54, No. 2, pp. 286-295.

Piaget, J. (1962). Play, dreams, and imitation in childhood. *New York : Norton*, 1962.

Popic, P. (2012). Analyse Statistique de résultats de simulations issus de la plateforme SocLab. Mémoire de stage L3 SID, Université Paul Sabatier Toulouse III, 2012.

Portet, F., Quiniou, R., Carrault, G., Cordier, M. O. (2008). Apprentissage d'arbre de décision pour le pilotage en ligne d'algorithmes de détection sur les électrocardiogrammes. *Actes de la 16e conférence Reconnaissance des Formes et Intelligence Artificielle (RFIA08)*, Amiens.

Price, B., Boutilier, C. (2003). Accelerating reinforcement learning through implicit imitation. *J. Artif. Intell. Res.*, vol 19, p. 569-629.

Riedmiller, M. A., Moore, A. W., Schneider, J. G. (2000). Reinforcement learning for cooperating and communicating reactive agents in electrical power grids. In *Balancing Reactivity and Social Deliberation in Multi-Agent Systems*, Hannebauer, M., Wendler, J., Pagello, E., Eds. New York : Springer, 2000, p. 137-149.

Robinet, V., Bisson, G., Gordon, M., Lemaire, B. (2008). Modèle cognitif de l'apprentissage inductif de concepts. In *Actes du colloque annuel de l'association pour la recherche cognitive* ARCo'08, 22-28. Lyon, France.

Roggero, P., Sibertin-Blanc, C. (2008). Quand des sociologues rencontrent des informaticiens : essai de formalisations des systèmes d'action concrets. *Nouvelles perspectives en sciences sociales*, 3(2) : 41-81.

Roggero, P., Vautier, C. (2003). L'opacité du système politico-institutionnel français : essai de modélisation complexe. *Res-Systemica*, 2003.

Ross, S. (1983). Introduction to Stochastic Dynamic Programming. *Academic Press*, New York.

Rummery, G. A., Niranjan, M. (1994). On-line Q-learning using connectionist systems. *Technical Report CUED/F-INFENG/TR 166*, Cambridge University Engineering Department.

Sacks, O. (1988). L'homme qui prenait sa femme pour un chapeau. *Essais 245, Seuil*, 1988.

Samuel, A. L. (1967). Some studies in machine learning using the game of checkers. II - Recent progress. *IBM Journal on Research and Development* : 601-617.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal on Research and Development*, pages 210-229. Reprinted in E. A. Feigenbaum, J. Feldman, editors, Computer and Thought. McGraw-Hill, New York, 1963.

Schaerf, A., Shoham, Y., Tennenholtz, M. (1995). Adaptive load balancing : A study in multi-agent learning. *J. Artif. Intell. Res.*, vol. 2, p. 475-500, 1995.

Schuster S. (2009). An algorithm for the Simulation of Bounded Rationel Agents. *MPRA Paper 15942*, University Library of Munich, Germany, 2009.

Scieur, P. (2005). Sociologie des organisations, Cursus. *Sociologie*. Colin, Armand.

Selten, R. (1998). Aspiration adaptation theory, *Journal of Mathematical Phychology*, 42, p. 191-214, 1998.

Shoham, Y., Powers, R., Grenager, T. (2007). If Multi-Agent Learning is the Answer, What is the Question ? *Artificial Intelligence* 171(7), p. 365-377, special issue on Foundations of Multi-Agent Learning (R. Vohra and M. Wellman, eds.).

Sibertin-Blanc, C., Roggero, P., Adreit, F., Baldet, B., Chapron, P., El Gemayel, J., Mailliard, M., Sandri, S. (2013). SocLab : A Framework for the Modelling, Simulation and Analysis of Power in Social Organizations. Dans : *Journal of Artificial Societies and Social Simulation*, University of Surrey, UK, 2013 (to appear).

Sibertin-Blanc, C., Mailliard, M. (2006). Un modèle de rationalité orienté vers la coopération. Dans : *Journées Francophones sur la Planification, la Décision et l'Apprentissage*, Toulouse, 10 – 12 mai 2006, Frédérick Garcia, Gérard Verfaillie (Eds.), INRA-ONERA, p. 57-64, mai 2006.

Sibertin-Blanc, C., Amblard, F., Mailliard, M. (2005). A Coordination Framework based on the Sociology of Organized Action. Dans : *OOOP 2005, From Organizations to Organization Oriented Programming in MAS* ; Workshop within the AAMAS'05 Conference, Utrecht (Nd), p. 1-16, Juillet, 2005.

Sibertin-Blanc, C., Roggero, P. (2004). Une formalisation de la Sociologie de l'Action Organisée. Dans : *XVIIème congrès de l'Association International des Sociologues de Langue Française*, Tours, p. 29, juillet 2004.

Sigaud O., Gérard P. (2003). Apprentissage par renforcement indirect dans les systèmes de classeurs. *Actes des journées PDMIA*, 2003.

Simon, H. A. (1982). Models of bounded rationality : Behavioral economics and business organization, Vol. 1-3. Cambridge, MA, *The MIT Press*, 1982-1997.

Simon, H. A. (1955). A Behavioral Model of Rational Choice. *Quarterly Journal of Economics,* 69 (1), p. 99-118.

Spaan, M. T. J., Vlassis, N., Groen, F. C. A. (2002). High level coordination of agents based on multi-agent Markov decision processes with rôles. In *Proc. Workshop Coop. Robot., 2002 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS-02)*, Lausanne, Switzerland, p. 66-73, october 2002.

Stephan, V., Debes, K, Gross, H. M., Wintrich, F, Wintrich, H. (2000). A reinforcement learning based neural multi-agent system for control a combustion process. In *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Netw. (IJCNN-00)*, Como, Italy, Jul. 24-27, p. 6217-6222.

Sun R., Slusarz P., Terry C. (2005). The Interaction of the Explicit and the Implicit in Skill Learning : A Dual-Process Approach. *Psychological Review*, vol. 112, n° 1, 2005, p. 159-192.

Sutton, R. S., Barto, A. G. (1998). Reinforcement Learning : An Introduction. *MIT Press*, Cambridge, MA, A Bradford Book.

Sutton, R. S. (1996). Generalization in reinforcement learning : Successful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems, Proceedings of the 1995 Conference*, Touretzky, D. S., Mozer, M. C., Hasselmo, M. E. (Eds.), p. 1038-1044, Cambridge, MA, MIT Press.

Sutton, R. S. (1991). Planning by incremental dynamic programming. *Proceedings of the Eighth International Workshop of Machine Learning*, p. 353-357, Morgan Kaufmann.

Sutton, R. S. (1990). Integrated Architectures for Learning, Planning, and Reacting Based on Approximating Dynamic Programming. In *Proceedings of the Seventh International Conference on Machine Learning*, p. 216-224, Morgan Kaufmann.

Sutton, R. S., Barto, A. G. (1987). A temporal-difference model of classical conditioning. In *Proceedings of the Ninth Annual Conference of the Computer Science Society*, Erlbaum, Hillsdale, NJ.

Tan, M. (1993). Multi-agent reinforcement learning : Independent vs cooperative agents. In *proceedings 10th International Conference on Machine Learning (ICML-93)*, Amherst, OH, June 27-29, p. 330-337.

Touzet, C. F. (2000). Robot awareness in cooperative mobile robot learning. *Auton. Robots*, vol. 8, no. 1, p. 87-97, 2000.

Vlassis, N. (2003). A concise introduction to multiagent systems and distributed AI. *Fac. Sci. Unv. Amsterdam*, Amsterdam, The Netherlands, Tech. Rep. [online]. Available : http://www.science.uva.nl/~vlassis/ cimasdai/cimasdai.pdf

Watkins, C. J. C. H., Dayan, P. (1992). Q-Learning. *Machine Learning*, 8:279-292.

Watkins, C. J. C. H. (1989). Learning from Delayed Rewards. PhD thesis, *Cambridge University*, Cambridge, England.

Weiss, G. (1999). Multiagent Systems : A Modern Approach to Distributed Artificial Intelligence. *Cambridge, MA, MIT Press*.

Wellman, M. P., Greenwald, A. R., Stone, P., Wurman, P. R. (2003). The 2001 trading agent competition. *Electron. Markets*, vol. 13, no. 1, p. 4-12, 2003.

Werbos, P. J. (1977). Advanced forecasting methods for global crisis warning and models of intelligence. *General Systems Yearbook*, 22:25-38.

White, D. J. (1969). Dynamic Programming. *Holden-Day*, San Francisco.

Whittle, P. (1983). Optimization over Time, Volume 2. *Wiley*, NY.

Whittle, P. (1982). Optimization over Time, Volume 1. *Wiley*, NY.

Wiering, M. (2000). Multi-agent reinforcement learning for traffic light control. In *Proc. 17th Int. Conf. Mach. Learn. (ICML-00)*, Stanford Univ., Stanford, CA, Jun. 29-Jul. 2, p. 1151-1158.